# Package: wehoop (via r-universe)

October 18, 2026

**Title** Access Women's Basketball Play by Play Data

**Version** 3.1.0

**Description** A utility for working with women's basketball data. A scraping and aggregating interface for the WNBA Stats API <https://stats.wnba.com/> and ESPN's women's college basketball and WNBA statistics. It provides users with the capability to access the game play-by-plays, box scores, standings and results to analyze the data for themselves.

**License** MIT + file LICENSE

**URL** https://wehoop.sportsdataverse.org, https://github.com/sportsdataverse/wehoop

**BugReports** https://github.com/sportsdataverse/wehoop/issues

**Depends** R (>= 4.1.0)

**Imports** cli (>= 3.0.0), data.table (>= 1.15.0), dplyr (>= 1.0.0), httr2 (>= 1.0.4), janitor, jsonlite (>= 1.6), lifecycle, progressr (>= 0.6.0), purrr (>= 1.0.0), Rcpp (>= 1.0.7), RcppParallel (>= 5.1.4), rlang (>= 1.0.0), rvest (>= 1.0.0), stringi (>= 1.7.0), stringr (>= 1.4.0), tidyr (>= 1.2.0), withr

**Suggests** arrow (>= 6.0.0), cachem (>= 1.0.0), crayon (>= 1.3.4), curl (>= 4.3.0), DBI, DT, ggplot2, ggrepel, gh (>= 1.0.0), glmnet, glue (>= 1.4.0), knitr, Matrix, memoise (>= 2.0.0), piggyback (>= 0.1.2), rmarkdown (>= 2.6.0), RSQLite, stats, stringdist (>= 0.9.0), testthat (>= 3.1.9), tibble (>= 3.0), tictoc, usethis (>= 1.6.0)

**Config/roxygen2/version** 8.1.0

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**Config/pak/sysreqs** cmake libicu-dev libxml2-dev libssl-dev

**Repository** https://sportsdataverse.r-universe.dev

**Date/Publication** 2026-10-18 22:06:56 UTC

**RemoteUrl** https://github.com/sportsdataverse/wehoop

**RemoteRef** HEAD

**RemoteSha** b6d8d2aa1309b1adda3885e317e22759ac2e13f1

# Contents

.players_on_court . . . 12
atrium_box_score . . . 12
atrium_competition_player_stats . . . 14
atrium_competition_team_stats . . . 16
atrium_end_year . . . 18
atrium_fixture . . . 19
atrium_fixture_periods . . . 21
atrium_fixtures . . . 23
atrium_fixtures_ribbon . . . 26
atrium_leaders . . . 28
atrium_pbp . . . 30
atrium_player . . . 32
atrium_player_game_log . . . 34
atrium_player_seasons . . . 37
atrium_player_stats . . . 39
atrium_players . . . 41
atrium_season_id . . . 43
atrium_seasons . . . 44
atrium_shots . . . 46
atrium_standings . . . 48
atrium_standings_summary . . . 50
atrium_team . . . 52
atrium_team_box_score . . . 55
atrium_team_stats . . . 57
atrium_teams . . . 59
bart_wbb_game_schedule . . . 61
bart_wbb_ratings . . . 62
bref_wnba_player_stats . . . 63
bref_wnba_standings . . . 64
bref_wnba_team_stats . . . 65
check_status . . . 66
csv_from_url . . . 66
espn_basketball . . . 66
espn_basketball_player_core . . . 67
espn_wbb_athletes_index . . . 69
espn_wbb_award . . . 70
espn_wbb_calendar . . . 71
espn_wbb_coach . . . 73

espn_wbb_coach_record . . . 74
espn_wbb_coach_season . . . 75
espn_wbb_coaches . . . 76
espn_wbb_conferences . . . 77
espn_wbb_franchise . . . 78
espn_wbb_franchises . . . 80
espn_wbb_game_all . . . 81
espn_wbb_game_odds . . . 83
espn_wbb_game_official_detail . . . 85
espn_wbb_game_play . . . 86
espn_wbb_game_play_personnel . . . 87
espn_wbb_game_player_box . . . 88
espn_wbb_game_probabilities . . . 90
espn_wbb_game_propbets . . . 91
espn_wbb_game_team_linescores . . . 92
espn_wbb_game_team_roster_entry . . . 94
espn_wbb_injuries . . . 95
espn_wbb_leaders . . . 97
espn_wbb_news . . . 98
espn_wbb_player_awards . . . 99
espn_wbb_player_career_stats . . . 100
espn_wbb_player_eventlog_v2 . . . 101
espn_wbb_player_info . . . 103
espn_wbb_player_overview . . . 105
espn_wbb_player_seasons . . . 108
espn_wbb_player_stats . . . 109
espn_wbb_position . . . 110
espn_wbb_positions . . . 111
espn_wbb_powerindex . . . 112
espn_wbb_rankings . . . 113
espn_wbb_scoreboard . . . 114
espn_wbb_season_awards . . . 115
espn_wbb_season_group . . . 117
espn_wbb_season_groups . . . 119
espn_wbb_season_info . . . 120
espn_wbb_season_leaders . . . 122
espn_wbb_season_ranking . . . 123
espn_wbb_season_rankings . . . 124
espn_wbb_season_type . . . 126
espn_wbb_season_types . . . 127
espn_wbb_season_week . . . 128
espn_wbb_season_weeks . . . 129
espn_wbb_seasons . . . 131
espn_wbb_standings . . . 132
espn_wbb_team . . . 133
espn_wbb_team_injuries . . . 135
espn_wbb_team_news . . . 136
espn_wbb_team_record_detail . . . 138

espn_wbb_team_schedule . . . 139
espn_wbb_team_season_roster . . . 140
espn_wbb_team_season_statistics . . . 142
espn_wbb_team_stats . . . 143
espn_wbb_teams . . . 144
espn_wbb_tournament . . . 145
espn_wbb_tournament_season . . . 146
espn_wbb_tournament_seasons . . . 147
espn_wbb_tournaments . . . 149
espn_wbb_venues . . . 150
espn_wbb_week_ranking . . . 151
espn_wbb_week_rankings . . . 152
espn_wnba_athletes_index . . . 154
espn_wnba_award . . . 155
espn_wnba_calendar . . . 156
espn_wnba_coach_season . . . 157
espn_wnba_coaches . . . 159
espn_wnba_conferences . . . 160
espn_wnba_draft . . . 161
espn_wnba_draft_athlete_detail . . . 162
espn_wnba_draft_pick . . . 164
espn_wnba_draft_rounds . . . 165
espn_wnba_franchise . . . 167
espn_wnba_franchises . . . 168
espn_wnba_freeagents . . . 169
espn_wnba_futures . . . 170
espn_wnba_game_all . . . 171
espn_wnba_game_odds . . . 173
espn_wnba_game_official_detail . . . 175
espn_wnba_game_play . . . 176
espn_wnba_game_play_personnel . . . 178
espn_wnba_game_player_box . . . 179
espn_wnba_game_probabilities . . . 180
espn_wnba_game_propbets . . . 181
espn_wnba_game_team_linescores . . . 183
espn_wnba_game_team_roster_entry . . . 185
espn_wnba_injuries . . . 186
espn_wnba_leaders . . . 187
espn_wnba_news . . . 188
espn_wnba_player_awards . . . 190
espn_wnba_player_career_stats . . . 191
espn_wnba_player_eventlog_v2 . . . 192
espn_wnba_player_info . . . 193
espn_wnba_player_overview . . . 195
espn_wnba_player_seasons . . . 198
espn_wnba_player_stats . . . 200
espn_wnba_position . . . 201
espn_wnba_positions . . . 202

espn_wnba_powerindex . . . 203
espn_wnba_scoreboard . . . 204
espn_wnba_season_awards . . . 205
espn_wnba_season_group . . . 207
espn_wnba_season_groups . . . 209
espn_wnba_season_info . . . 210
espn_wnba_season_leaders . . . 212
espn_wnba_season_ranking . . . 213
espn_wnba_season_rankings . . . 215
espn_wnba_season_type . . . 216
espn_wnba_season_types . . . 217
espn_wnba_season_week . . . 218
espn_wnba_season_weeks . . . 220
espn_wnba_seasons . . . 221
espn_wnba_standings . . . 222
espn_wnba_team . . . 224
espn_wnba_team_injuries . . . 226
espn_wnba_team_news . . . 227
espn_wnba_team_record . . . 228
espn_wnba_team_record_detail . . . 230
espn_wnba_team_schedule . . . 231
espn_wnba_team_season_roster . . . 232
espn_wnba_team_season_statistics . . . 234
espn_wnba_team_stats . . . 235
espn_wnba_teams . . . 236
espn_wnba_transactions . . . 237
espn_wnba_venues . . . 238
espn_wnba_week_ranking . . . 240
espn_wnba_week_rankings . . . 241
feb_game_info . . . 243
feb_leaders . . . 245
feb_overview . . . 246
feb_pbp . . . 248
feb_period_scores . . . 250
feb_player_box . . . 251
feb_results . . . 253
feb_schedule . . . 255
feb_shots . . . 257
feb_standings . . . 259
feb_team_stats . . . 261
fiba_ecw_competition . . . 263
fiba_elw_competition . . . 271
fiba_gamepage_box_players . . . 280
fiba_gamepage_box_teams . . . 283
fiba_gamepage_lineup . . . 286
fiba_gamepage_pbp . . . 288
fiba_gamepage_period_scores . . . 290
fiba_gamepage_shots . . . 292

fiba_gdap_awards . . . 294
fiba_gdap_competition . . . 296
fiba_gdap_editions . . . 298
fiba_gdap_federation_games . . . 301
fiba_gdap_federation_ranking . . . 304
fiba_gdap_final_standings . . . 306
fiba_gdap_game . . . 307
fiba_gdap_game_officials . . . 310
fiba_gdap_game_periods . . . 311
fiba_gdap_games_between . . . 313
fiba_gdap_gateway_competition_ids . . . 316
fiba_gdap_gateway_federation_ranking . . . 318
fiba_gdap_medal_count . . . 320
fiba_gdap_national_teams . . . 321
fiba_gdap_nominations . . . 323
fiba_gdap_player_game_stats . . . 325
fiba_gdap_player_info . . . 327
fiba_gdap_player_stats . . . 329
fiba_gdap_ranking_zones . . . 332
fiba_gdap_roster . . . 333
fiba_gdap_roster_coaches . . . 335
fiba_gdap_roster_latest . . . 337
fiba_gdap_round_gamedays . . . 339
fiba_gdap_round_standings . . . 341
fiba_gdap_rounds . . . 343
fiba_gdap_schedule . . . 345
fiba_gdap_team_games . . . 348
fiba_gdap_team_player_stats . . . 351
fiba_gdap_team_staff . . . 354
fiba_gdap_team_stats . . . 356
fiba_gdap_teams . . . 359
fiba_wafrobasket_competition . . . 361
fiba_wamericup_competition . . . 370
fiba_wasiacup_competition . . . 378
fiba_weurobasket_competition . . . 387
fiba_woly_competition . . . 395
fiba_wwc_competition . . . 404
fox_basketball_league_leaders . . . 412
fox_wbb_teams_all . . . 413
fox_wnba_pbp . . . 414
fox_wnba_team_roster . . . 415
genius_competition_games . . . 417
genius_competitions . . . 419
genius_game . . . 421
genius_game_crosswalk . . . 422
genius_match_competition . . . 424
genius_pbp . . . 425
genius_player_box . . . 427

genius_schedule . . . 430
genius_schedule_embed . . . 432
genius_shots . . . 433
genius_team_box . . . 435
genius_team_crosswalk . . . 438
genius_widget_games . . . 439
helper_espn_wbb_pbp . . . 440
helper_espn_wnba_pbp . . . 441
hhs_team_roster . . . 442
hhs_team_stats . . . 443
hhs_teams . . . 444
lfb_competition . . . 445
lmb_competition . . . 451
lnbp_competition . . . 457
load_ncaa_wbb_pbp . . . 464
load_wbb_group_aliases . . . 470
load_wbb_group_seasons . . . 471
load_wbb_groups . . . 472
load_wbb_pbp . . . 473
load_wbb_team_group_seasons . . . 478
load_wnba_pbp . . . 480
load_wnba_player_impact . . . 485
load_wnba_stats_leaguedash . . . 487
load_wnba_stats_rosters . . . 488
load_wnba_team_crosswalk . . . 495
lpb_competition . . . 497
most_recent_wbb_season . . . 504
most_recent_wnba_season . . . 504
most_recent_wnba_stats_season . . . 505
nbl_rosetta_match . . . 505
nbl_rosetta_match_by_id . . . 508
nbl_rosetta_match_pbp . . . 510
nbl_rosetta_match_player_box . . . 512
nbl_rosetta_match_player_period_box . . . 515
nbl_rosetta_match_team_box . . . 517
nbl_rosetta_matches . . . 519
nbl_rosetta_next_match . . . 523
nbl_rosetta_next_match_team . . . 525
nbl_rosetta_next_matches . . . 528
nbl_rosetta_outcomes . . . 530
nbl_rosetta_player_game_log . . . 532
nbl_rosetta_player_stats . . . 535
nbl_rosetta_players . . . 539
nbl_rosetta_seasons . . . 541
nbl_rosetta_seasons_current . . . 543
nbl_rosetta_standings . . . 545
nbl_rosetta_stats_leaders . . . 547
nbl_rosetta_team_players . . . 550

nbl_rosetta_team_stats . . . 552
nbl_rosetta_team_stats_leaders . . . 555
nbl_rosetta_teams . . . 558
nbl1_competition . . . 560
ncaa_wbb . . . 567
ncaa_wbb_NET_rankings . . . 568
ncaa_wbb_teams . . . 568
parameter_descriptions . . . 569
parquet_from_url . . . 570
progressively . . . 570
rds_from_url . . . 571
scb_game . . . 571
scb_game_actions . . . 575
scb_game_players . . . 577
scb_game_teams . . . 579
scb_games . . . 582
scb_games_recent . . . 586
scb_player_statistics . . . 589
scb_playoff_games . . . 592
scb_playoff_series . . . 594
scb_records_per_game . . . 596
scb_records_season_averages . . . 598
scb_referees . . . 599
scb_season_player . . . 601
scb_season_players . . . 603
scb_season_teams . . . 604
scb_seasons . . . 606
scb_standings . . . 608
scb_team . . . 610
scb_team_rosters . . . 612
scb_team_seasons . . . 614
scb_team_statistics . . . 615
scb_transfers . . . 618
scb_transfers_teams . . . 620
tbf_daily_matches . . . 621
tbf_game_header . . . 623
tbf_game_leaders . . . 625
tbf_game_players . . . 627
tbf_highlights . . . 628
tbf_leaders_assists . . . 630
tbf_leaders_blocks . . . 632
tbf_leaders_efficiency . . . 633
tbf_leaders_minutes . . . 635
tbf_leaders_points . . . 637
tbf_leaders_rebounds . . . 638
tbf_leaders_steals . . . 640
tbf_leaders_threes . . . 642
tbf_leaders_turnovers . . . 643

tbf_league_seasons . . . 645
tbf_leagues . . . 647
tbf_pbp . . . 648
tbf_period_scores . . . 650
tbf_player_box . . . 651
tbf_player_career . . . 654
tbf_players . . . 656
tbf_schedule . . . 657
tbf_seasons . . . 659
tbf_shots . . . 661
tbf_standings . . . 662
tbf_team_box . . . 664
tbf_team_games . . . 666
tbf_team_player_stats . . . 668
tbf_team_roster . . . 671
tbf_team_staff . . . 673
tbf_teams . . . 674
tbf_weeks . . . 676
update_wbb_db . . . 678
update_wnba_db . . . 679
update_wnba_stats_db . . . 681
wbb_player_crosswalk . . . 682
wbb_schedule_crosswalk . . . 682
wbb_team_crosswalk . . . 683
wbbl_competition . . . 684
wehoop_loaders . . . 690
wjbl_box_score . . . 691
wjbl_game . . . 693
wjbl_game_info . . . 695
wjbl_game_months . . . 698
wjbl_head_to_head . . . 700
wjbl_leagues . . . 702
wjbl_pbp . . . 704
wjbl_period_scores . . . 706
wjbl_player . . . 708
wjbl_player_period_leaders . . . 710
wjbl_roster . . . 712
wjbl_schedule . . . 714
wjbl_staff . . . 717
wjbl_standing_seasons . . . 719
wjbl_team_box_score . . . 721
wjbl_team_period_stats . . . 723
wjbl_team_ranking . . . 725
wjbl_team_standings . . . 728
wjbl_teams . . . 730
wnba_alltimeleadersgrids . . . 731
wnba_assistleaders . . . 736
wnba_assisttracker . . . 738

wnba_boxscoresummaryv3 . . . 739
wnba_boxscoretraditionalv2 . . . 740
wnba_boxscoretraditionalv3 . . . 743
wnba_commonallplayers . . . 748
wnba_commonplayerinfo . . . 749
wnba_commonplayoffseries . . . 750
wnba_commonteamroster . . . 751
wnba_cumestatsplayer . . . 752
wnba_cumestatsplayergames . . . 753
wnba_cumestatsteam . . . 754
wnba_cumestatsteamgames . . . 755
wnba_data_pbp . . . 757
wnba_draftboard . . . 758
wnba_draftcombinestats . . . 759
wnba_drafthistory . . . 761
wnba_fantasywidget . . . 762
wnba_franchisehistory . . . 764
wnba_franchiseleaders . . . 765
wnba_franchiseleaderswrank . . . 766
wnba_franchiseplayers . . . 767
wnba_gamerotation . . . 768
wnba_homepagewidget . . . 769
wnba_infographicfanduelplayer . . . 771
wnba_leaguedashlineups . . . 772
wnba_leaguedashplayerbiostats . . . 774
wnba_leaguedashplayerclutch . . . 777
wnba_leaguedashplayershotlocations . . . 780
wnba_leaguedashplayerstats . . . 783
wnba_leaguedashteamclutch . . . 786
wnba_leaguedashteamshotlocations . . . 788
wnba_leaguedashteamstats . . . 791
wnba_leaguegamefinder . . . 793
wnba_leaguegamelog . . . 798
wnba_leagueleaders . . . 800
wnba_leaguelineupviz . . . 801
wnba_leagueplayerondetails . . . 803
wnba_leagueseasonmatchups . . . 805
wnba_leaguestandingsv3 . . . 806
wnba_live_boxscore . . . 808
wnba_live_pbp . . . 809
wnba_pbp . . . 810
wnba_pbps . . . 811
wnba_playbyplayv3 . . . 812
wnba_player_crosswalk . . . 813
wnba_playerawards . . . 813
wnba_playercareerbycollege . . . 814
wnba_playercareerbycollegerollup . . . 816
wnba_playercareerstats . . . 817

wnba_playercompare . . . . . 819
wnba_playerdashboardbyclutch . . . . . 821
wnba_playerestimatedmetrics . . . . . 834
wnba_playerfantasyprofile . . . . . 835
wnba_playerfantasyprofilebargraph . . . . . 837
wnba_playergamelog . . . . . 838
wnba_playergamelogs . . . . . 839
wnba_playergamestreakfinder . . . . . 841
wnba_playerheadshot . . . . . 846
wnba_playerindex . . . . . 847
wnba_playernextngames . . . . . 849
wnba_playerprofilev2 . . . . . 850
wnba_playervsplayer . . . . . 853
wnba_possession_lineups . . . . . 856
wnba_rapm . . . . . 856
wnba_referee_assignments . . . . . 857
wnba_schedule . . . . . 859
wnba_schedule_crosswalk . . . . . 860
wnba_scoreboardv2 . . . . . 860
wnba_scoreboardv3 . . . . . 863
wnba_shot_zones . . . . . 864
wnba_shotchartdetail . . . . . 866
wnba_shotchartleaguewide . . . . . 868
wnba_shotchartlineupdetail . . . . . 869
wnba_stats . . . . . 873
wnba_team_crosswalk . . . . . 874
wnba_teamdashboardbyclutch . . . . . 875
wnba_teamdashlineups . . . . . 886
wnba_teamdetails . . . . . 889
wnba_teamestimatedmetrics . . . . . 891
wnba_teamgamelog . . . . . 892
wnba_teamgamelogs . . . . . 893
wnba_teamgamestreakfinder . . . . . 895
wnba_teamhistoricalleaders . . . . . 905
wnba_teaminfocommon . . . . . 906
wnba_teamplayeronoffdetails . . . . . 907
wnba_teams . . . . . 911
wnba_teamvsplayer . . . . . 912
wnba_teamyearbyyearstats . . . . . 918
wnba_todays_scoreboard . . . . . 919
wnba_videoevents . . . . . 920
wnba_videostatus . . . . . 921
year_to_season . . . . . 922

**Index** **923**

## .players_on_court — Add players on court in WNBA Stats API play-by-play

### Description

**Add players on court in WNBA Stats API play-by-play**

## atrium_box_score — Atrium Player Box Score

### Description

**One game's box score: one row per rostered player with the statistics the site configures.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=s`

A player who did not play has `participated` `FALSE` and `NA` statistics. LNB has `blocks_received` and `plus_minus` but no field-goal totals; LNBP the reverse. `minutes` is the ISO duration as sent and `seconds_played` its value. Percentages are 0-100. `side` is `home` or `away`.

### Usage

```
atrium_box_score(website_id, fixture_id, locale = NULL)
```

### Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in `atrium_fixtures()`), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

### Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(fixture_id, person_id)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID the player is rostered with for this game. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| person_name | character | Player name as the site prints it (upper case on LNBP and NBL1). |
| bib | character | Jersey number the player wears, as text. |
| position | character | Position code on the roster (LNB, e.g. PG); null on LNBP. |
| starter | logical | True when the player started the game. |
| active | logical | True when the player is on the court in a live game. |
| participated | logical | True when the player entered the game; false rows have null statistics. |
| did_not_play_reason | character | Reason the player did not play; null in every body seen. |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, this game). |
| points | integer | Points scored (player, this game). |
| rebounds | integer | Total rebounds (player, this game). |
| rebounds_offensive | integer | Offensive rebounds (player, this game). |
| rebounds_defensive | integer | Defensive rebounds (player, this game). |
| assists | integer | Assists (player, this game). |
| steals | integer | Steals (player, this game). |
| blocks | integer | Shots blocked (player, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, this game). |
| turnovers | integer | Turnovers (player, this game). |
| fouls_total | integer | Fouls committed (player, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, this game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) ( |
| efficiency | integer | Efficiency rating, from the site's own formula (player, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, this game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, this game). |

| | | |
|---|---|---|
| points_two_made | integer | Two-point field goals made (player, this game). |
| points_two_attempted | integer | Two-point field goals attempted (player, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, this game). |
| points_three_made | integer | Three-point field goals made (player, this game). |
| points_three_attempted | integer | Three-point field goals attempted (player, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, this game). |
| free_throws_made | integer | Free throws made (player, this game). |
| free_throws_attempted | integer | Free throws attempted (player, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, this game). |

## See Also

Other Atrium Functions: atrium_competition_player_stats(), atrium_competition_team_stats(), atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon, atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(), atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

## Examples

```
try(atrium_box_score(website_id = 14, fixture_id = "0dc631bd-c34c-11f1-8d16-4d5a21afb79d"))
```

---

`atrium_competition_player_stats`

# Atrium Competition Player Stats (long form)

---

## Description

**Player statistics of a competition in long form (names only: the bodies carry no ids).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/statistics_competiti`

## Usage

```
atrium_competition_player_stats(website_id, competition_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| competition_id | (*character* required): Competition UUID (`competition_id` in atrium_seasons()), e.g. `"3f4064bb-51ad-11f0-aaaf-2923c944b404"`. |
| locale | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_competition_player_stats(
  website_id = 12, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
)
```

## Value

A `wehoop_data` tibble, one row per `(table_key, person_name, stat_key)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 |
| season_id | character | Season UUID of the table; null for the competition-scoped routes (their bodies carry none). |
| table_key | character | Statistics table the row is from: averages, shootingstatistics, totals, ... (the set is per site). |
| table_label | character | The table's heading in the site's language (e.g. Moyennes). |
| person_id | character | Player UUID decoded from the widget link beside the name (the tables carry names only); null wl |
| person_name | character | Player name as the site prints it; null in team tables. |
| entity_id | character | Team UUID decoded from the widget link (team tables only; player tables name the team but carr |
| entity_name | character | Team name as the site prints it; for player rows of a stat table it is the player's team. |
| stat_key | character | Statistic key as the API names it (pointsPerGame, fieldGoalsMade, ...), unchanged from header ke |
| stat_name | character | Short column heading the site shows for the statistic (PTS, REB), in the site's language. |
| stat_label | character | Longer statistic label the site shows (code in the API), in the site's language. |
| value | double | Numeric value of the statistic; null when the API sends a duration or text (see value_text). |
| value_text | character | Non-numeric statistic value, e.g. an ISO 8601 duration such as PT12M2S for minutes per game. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_team_stats(), atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon(), atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(), atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

`atrium_competition_team_stats`

# Atrium Competition Team Stats (long form)

---

## Description

**Team statistics of a competition in long form (names only: the bodies carry no ids).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/statistics_competiti`

## Usage

```
atrium_competition_team_stats(website_id, competition_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `competition_id` | (*character* required): Competition UUID (`competition_id` in `atrium_seasons()`), e.g. `"3f4064bb-51ad-11f0-aaaf-2923c944b404"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across

sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_competition_team_stats(
  website_id = 12, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
)
```

## Value

A `wehoop_data` tibble, one row per `(table_key, entity_name, stat_key)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 A |
| season_id | character | Season UUID of the table; null for the competition-scoped routes (their bodies carry none). |
| table_key | character | Statistics table the row is from: averages, shootingstatistics, totals, ... (the set is per site). |
| table_label | character | The table's heading in the site's language (e.g. Moyennes). |
| person_id | character | Player UUID decoded from the widget link beside the name (the tables carry names only); null wl |
| person_name | character | Player name as the site prints it; null in team tables. |
| entity_id | character | Team UUID decoded from the widget link (team tables only; player tables name the team but carr |
| entity_name | character | Team name as the site prints it; for player rows of a stat table it is the player's team. |
| stat_key | character | Statistic key as the API names it (pointsPerGame, fieldGoalsMade, ...), unchanged from header ke |
| stat_name | character | Short column heading the site shows for the statistic (PTS, REB), in the site's language. |
| stat_label | character | Longer statistic label the site shows (code in the API), in the site's language. |
| value | double | Numeric value of the statistic; null when the API sends a duration or text (see value_text). |
| value_text | character | Non-numeric statistic value, e.g. an ISO 8601 duration such as PT12M2S for minutes per game. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon()`, `atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`, `atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`, `atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`, `atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

`atrium_end_year` **Atrium END Year**

## Description

**The sdv season key (END year) of a site's own `year`, or `NA` when it cannot be mapped.**

## Usage

```
atrium_end_year(website_id, year)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): The site (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). |
| `year` | (*integer* required): The `year` of a site's season list (START year on site 12, the calendar year elsewhere); a vector is mapped element by element. |

## Value

An integer vector: `year` + 1 on site 12 (2025 -> 2026 = 2025-26), `year` on the calendar-year sites, `NA` for an unregistered site, site 236 (no season list) or a value that is not a whole number.

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon()`,
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_team_stats()`, `atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`,
`nbl1_competition`

## Examples

```
atrium_end_year(12, 2025)
atrium_end_year(14, 2026)
```

`atrium_fixture` **Atrium Fixture (game header)**

## Description

**One game's header: teams, score, venue, status, rules (periods, overtime, shot clock) and captains.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=s`

The rules (`number_of_periods`, `period_length` in minutes, `overtime_length`, `shot_clock` in seconds, ...) come from the game's profile. `*_captain_person_id` is the person id of each team's captain.

## Usage

```
atrium_fixture(website_id, fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in `atrium_fixtures()`), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument

that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row for the game:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes |
| season_id | character | Season UUID of the game. |
| competition_name | character | Name of the competition, in the site's language. |
| name | character | Fixture name; null on LNBP. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| age_group | character | Age group of the competition, e.g. SENIOR. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (`YYYY-MM-DDTHH:MM:` |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_ |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| external_id | character | The league's own match number (LNB: the lnb.fr number); null elsewhere. |
| updated | POSIXct | When the record was last updated, naive datetime (no offset; zone unverified). |
| live_data_available | logical | True when live data feeds are enabled for the game. |
| live_video_available | logical | True when a live video stream is attached. |
| discipline | character | Discipline label of the game; null in every body seen. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside |
| number_of_periods | integer | Regulation periods in the game (4). |
| period_length | integer | Length of a regulation period, in minutes. |
| overtime_allowed | logical | True when the game may go to overtime. |
| overtime_length | integer | Length of an overtime period, in minutes. |
| shot_clock | integer | Shot clock length, in seconds. |
| max_fouls_personal | integer | Personal fouls that disqualify a player. |
| fouls_before_bonus | integer | Team fouls in a period before the bonus applies. |
| maximum_players_on_court | integer | Players a team has on court at once. |
| maximum_roster_size | integer | Largest roster a team may enter for the game. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |

| | | |
|---|---|---|
| home_captain_person_id | character | Person UUID of the home team's captain; null when the body names none. |
| away_captain_person_id | character | Person UUID of the away team's captain; null when the body names none. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon(),
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(),
atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition,
nbl1_competition

## Examples

```
try(atrium_fixture(website_id = 14, fixture_id = "0dc631bd-c34c-11f1-8d16-4d5a21afb79d"))
```

---

`atrium_fixture_periods`

# Atrium Fixture Periods

---

## Description

**One game's periods with the points each team scored in each, overtime periods included (ids 11+).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=s`

`home_points` / `away_points` are the points scored IN that period, not the running score. Overtime periods are `period_id` 11, 12, ... (not 5).

## Usage

```
atrium_fixture_periods(website_id, fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in `atrium_fixtures()`), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_fixture_periods(website_id = 14, fixture_id = "0dc631bd-c34c-11f1-8d16-4d5a21afb79d")
```

## Value

A `wehoop_data` tibble, one row per `(fixture_id, period_id)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail rou |
| season_id | character | Season UUID of the game. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| label | character | Short period label in the site's language (Q1, P1), as sent. |
| is_overtime | logical | True for an overtime period. |
| is_shootout | logical | True for a shootout period. |
| started | logical | True once the period has started. |
| ended | logical | True once the period has ended. |
| duration_minutes | integer | Length of the period in minutes. |
| elapsed_minutes_before_period | integer | Minutes of game time played before the period starts (40 for the first overtime). |
| home_entity_id | character | Team UUID of the home team, from the fixture's isHome flag. |
| away_entity_id | character | Team UUID of the away team, from the fixture's isHome flag. |
| home_points | integer | Points the home team scored in this period (not the running score). |
| away_points | integer | Points the away team scored in this period (not the running score). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixtures(), atrium_fixtures_ribbon(), atrium_leaders(),
atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(), atrium_player_stats(),
atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

`atrium_fixtures` **Atrium Fixtures (schedule and results)**

---

## Description

**Fixtures (schedule and results) of a season, home and away side by side.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixtures`

Scores are integers (the API sends strings); `start_time_utc` is a UTC `POSIXct`; `start_time_local` is the venue's wall-clock time as the API sends it (character: its zone, `timezone`, varies per row, so no single `POSIXct` zone fits). sdv-py keeps both naive.

## Usage

```
atrium_fixtures(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  entity_id = NULL,
  round_number = NULL,
  sub = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `season` | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| `competition_id` | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |

| | |
|---|---|
| `entity_id` | (*character* optional): Team UUID (`entity_id` in `atrium_teams()` or any fixture row). |
| `round_number` | (*integer* optional): Round (journee) number to show; the pages that take it list their rounds. |
| `sub` | (*character* optional): Tab: `"FIXTURES"` (upcoming) or `"RESULTS"` (played). Omitted = the site's default tab. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `fixture_id`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| season_id | character | Season UUID the page is about (the site's default season when none was asked). |
| competition_id | character | Competition UUID of the season. |
| competition_name | character | Competition name in the site's language, e.g. Betclic ELITE or LIGA CALIENTE.MX |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| name | character | Fixture name as the site prints it (e.g. Paris - Lyon-Villeurbanne); null on LNBP. |
| round | character | Round of the game as the site words it, e.g. Journee: 30. |
| round_code | character | Round code; null in every body seen. |

| | | |
|---|---|---|
| pool | character | Pool the game belongs to; null when the competition has none. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a s |
| stage_code | character | Opaque 30-hex stage code of the game; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (YYYY-MM-DDTHH:MM:SS, |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_loc |
| times_unconfirmed | logical | True while the start time is not final. |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_place | integer | Result place of the home team: 1 for the winner, 2 for the loser. |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_draw | logical | True when the game ended level for the home team. |
| home_shootout_attempts | character | Shootout attempts of the home team as sent (an empty string becomes null). |
| home_secondary_score | character | Secondary score of the home team (unused in the samples; shown as text). |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_place | integer | Result place of the away team: 1 for the winner, 2 for the loser. |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_draw | logical | True when the game ended level for the away team. |
| away_shootout_attempts | character | Shootout attempts of the away team as sent (an empty string becomes null). |
| away_secondary_score | character | Secondary score of the away team (unused in the samples; shown as text). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures_ribbon(),
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(),
atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition,
nbl1_competition

## Examples

```
try(atrium_fixtures(website_id = 14))
```

`atrium_fixtures_ribbon`

**Atrium Fixtures Ribbon**

## Description

**The fixtures ribbon: recent and upcoming games across the site's competitions (empty on LNB off-season).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixtures_ribbon`

Its scores are integers; `start_time_local` is the ribbon's `fixture.date` (a local time, no offset; unverified).

## Usage

```
atrium_fixtures_ribbon(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  entity_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `season` | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| `competition_id` | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |
| `entity_id` | (*character* optional): Team UUID (`entity_id` in `atrium_teams()` or any fixture row). |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_fixtures_ribbon(website_id = 14)
```

## Value

A `wehoop_data` tibble, one row per `fixture_id`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama |
| season_id | character | Season UUID of the game. |
| season_name | character | Name of the season / competition the game belongs to. |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a series. |
| stage_name | character | Name of the stage; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| is_final | logical | True when the game is over. |
| is_live | logical | True while the game is being played. |
| times_unconfirmed | logical | True while the start time is not final. |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | The ribbon's fixture date as the API sends it: a local wall-clock time (no offset; the zone is r |
| home_entity_id | character | Team UUID of the home team. |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_is_leading | logical | True while the home team leads (ribbon rows). |

| | | |
|---|---|---|
| away_entity_id | character | Team UUID of the away team. |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_is_leading | logical | True while the away team leads (ribbon rows). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_leaders(),
atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(), atrium_player_stats(),
atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

`atrium_leaders` **Atrium Leaders**

---

## Description

**Season leaders per statistic (top 5 each), with their player and team ids.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/statistics_leaders`

`rank` is the 1-based position within the statistic. `value` is numeric; a duration statistic (`minutesPerGame`) lands in `value_text` as an ISO string.

## Usage

```
atrium_leaders(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  entity_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see atrium_seasons()). Omit `season` and `season_id` for the site's default season. |

`season_id` (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`.

`competition_id` (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it.

`entity_id` (*character* optional): Team UUID (`entity_id` in `atrium_teams()` or any fixture row).

`locale` (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change.

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_leaders(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
)
```

## Value

A `wehoop_data` tibble, one row per `(season_id, stat_key, rank)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 A |
| season_id | character | Season UUID of the leaders. |
| stat_key | character | Statistic the leader is ranked in (points, pointsPerGame, efficiency, doubleDouble, ...). |
| stat_name | character | Short column heading the site shows for the statistic (PTS, REB), in the site's language. |
| stat_label | character | Longer statistic label the site shows (code in the API), in the site's language. |

| | | |
|---|---|---|
| rank | integer | Position of the leader within the statistic, 1-based (the API lists the top 5). |
| entity_id | character | Team UUID of the leader's team. |
| entity_name | character | Name of the leader's team. |
| person_id | character | Player UUID of the leader. |
| person_name | character | Name of the leader. |
| value | double | Numeric value of the statistic; null when the API sends a duration or text (see value_text). |
| value_text | character | Non-numeric statistic value, e.g. an ISO 8601 duration such as PT12M2S for minutes per game. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(), atrium_player_stats(),
atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

`atrium_pbp` **Atrium Play-by-Play**

---

## Description

**One game's play-by-play: every event with the period clock, the player and the running score.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=p`

`sequence` is the 1-based position in the body (periods in order, events chronological within a period). `clock` is the time REMAINING in the period as sent (`PT10M0S` at the start) and `clock_seconds` its value. `home_score` / `away_score` are the running score after the event. `x` / `y` (shots only) are percent of the court (0-100). Team events have no `person_id`. Overtime is `period_id` 11+.

## Usage

```
atrium_pbp(website_id, fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in atrium_fixtures()), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `event_id`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the event in the game (periods in order, events chronological). |
| event_id | character | Event UUID, shared with the same event in atrium_shots. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID; null for team events (time-outs, team rebounds). |
| person_name | character | Player the event is credited to; null for team events (time-outs, team rebounds). |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when a shot or free throw was made, false when missed; null for other events. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |

| | | |
|---|---|---|
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(),
atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition,
nbl1_competition

## Examples

```
try(atrium_pbp(website_id = 14, fixture_id = "0dc631bd-c34c-11f1-8d16-4d5a21afb79d"))
```

---

`atrium_player` **Atrium Player Page**

---

## Description

**A player page: bio and the season averages the card shows.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/person_detail`

Info items the documented columns do not list (height, position, ...) are appended as columns named after their key.

## Usage

```
atrium_player(
  website_id,
  person_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| person_id | (*character* required): Player UUID (`person_id` in `atrium_players()` or a box score), e.g. `"4458f165-f865-11f0-8ebf-539c931c1654"`. |

| | |
|---|---|
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs competition_id when several competitions share that year (see atrium_seasons()). Omit season and season_id for the site's default season. |
| season_id | (*character* optional): Season UUID (season_id in atrium_seasons()); sent as is, skipping the END-year lookup of season. |
| competition_id | (*character* optional): Competition UUID (competition_id in atrium_seasons()). Picks the season together with season, and is then not sent; with neither season nor season_id the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. "en-EN" or "fr-FR". Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** website_id picks the league site; one site carries many competitions. Seasons are END years (season = 2026): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair season with competition_id (see atrium_seasons()) or pass season_id. atrium_end_year() maps a site's own year to that END year, and atrium_season_id() finds the season UUID.

**Columns** are configured per site: a statistic one site omits is NA. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so person_id there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); wehoop_bad_request for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, season with season_id); wehoop_asset_fetch for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_player(website_id = 12, person_id = "4458f165-f865-11f0-8ebf-539c931c1654")
```

## Value

A wehoop_data tibble, one row for the player:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| name | character | Player name as the site prints it. |
| bib | character | Jersey number on the card; null when none. |
| team_id | character | Team UUID of the player's current team on the card. |

| | | |
|---|---|---|
| team_name | character | Name of the player's current team on the card. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |
| date_of_birth | Date | Player's date of birth, from the card's info items. |
| age | integer | Player's age in years, from the card's info items. |
| nationality | character | Player's nationality in the site's language, from the card's info items. |
| points_per_game | double | Points per game over the season shown on the card. |
| rebounds_offensive_per_game | double | Offensive rebounds per game over the season shown on the card. |
| rebounds_defensive_per_game | double | Defensive rebounds per game over the season shown on the card. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(),
atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, lpb_competition,
nbl1_competition

---

`atrium_player_game_log`

# Atrium Player Game Log

---

## Description

**A player's game log: one row per game with the opponent, result and the statistics the site shows.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/person_detail?sub=lo`

`score` is the site's text (`"88 - 91"`); which side is listed first is unverified, so it is not split. `date` is the local start time (no offset). The log carries no fixture id: join on date and opponent.

## Usage

```
atrium_player_game_log(
  website_id,
  person_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| person_id | (*character* required): Player UUID (person_id in atrium_players() or a box score), e.g. "4458f165-f865-11f0-8ebf-539c931c1654". |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs competition_id when several competitions share that year (see atrium_seasons()). Omit season and season_id for the site's default season. |
| season_id | (*character* optional): Season UUID (season_id in atrium_seasons()); sent as is, skipping the END-year lookup of season. |
| competition_id | (*character* optional): Competition UUID (competition_id in atrium_seasons()). Picks the season together with season, and is then not sent; with neither season nor season_id the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. "en-EN" or "fr-FR". Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** website_id picks the league site; one site carries many competitions. Seasons are END years (season = 2026): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair season with competition_id (see atrium_seasons()) or pass season_id. atrium_end_year() maps a site's own year to that END year, and atrium_season_id() finds the season UUID.

**Columns** are configured per site: a statistic one site omits is NA. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so person_id there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); wehoop_bad_request for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, season with season_id); wehoop_asset_fetch for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_player_game_log(website_id = 12, person_id = "4458f165-f865-11f0-8ebf-539c931c1654")
```

## Value

A wehoop_data tibble, one row per game:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Par |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| season_id | character | Season UUID the log is for (the season asked, else the site's default). |
| date | character | Local start of the game as the API sends it (YYYY-MM-DDTHH:MM:SS, no offset; the zone |
| opponent_entity_id | character | Team UUID of the opponent. |
| opponent_name | character | Name of the opponent. |
| result | character | Result for the player's team: W or L. |
| result_label | character | Result label in the site's language (V, D). |
| score | character | Final score as the site's text (e.g. 88 - 91); which side is listed first is unverified, so it is |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, that game). |
| points | integer | Points scored (player, that game). |
| rebounds | integer | Total rebounds (player, that game). |
| rebounds_offensive | integer | Offensive rebounds (player, that game). |
| rebounds_defensive | integer | Defensive rebounds (player, that game). |
| assists | integer | Assists (player, that game). |
| steals | integer | Steals (player, that game). |
| blocks | integer | Shots blocked (player, that game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, that game). |
| turnovers | integer | Turnovers (player, that game). |
| fouls_total | integer | Fouls committed (player, that game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, that game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) (p |
| efficiency | integer | Efficiency rating, from the site's own formula (player, that game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, that game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, that game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, that game). |
| points_two_made | integer | Two-point field goals made (player, that game). |
| points_two_attempted | integer | Two-point field goals attempted (player, that game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, that game). |
| points_three_made | integer | Three-point field goals made (player, that game). |
| points_three_attempted | integer | Three-point field goals attempted (player, that game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, that game). |
| free_throws_made | integer | Free throws made (player, that game). |
| free_throws_attempted | integer | Free throws attempted (player, that game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, that game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, that game). |
| dunks | integer | Dunks (player, that game). |
| fouls_personal | integer | Personal fouls (player, that game). |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_seasons()`, `atrium_player_stats()`,
`atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`, `atrium_standings()`,
`atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`,

atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

`atrium_player_seasons` **Atrium Player Seasons**

---

## Description

**A player's per-game averages, one row per competition-season.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/person_detail`

`season_year` is the site's own year and `season_end_year` the sdv END year. The statistics are the per-game averages the site configures (double rates, ISO `minutes_per_game`).

## Usage

```
atrium_player_seasons(
  website_id,
  person_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `person_id` | (*character* required): Player UUID (`person_id` in `atrium_players()` or a box score), e.g. `"4458f165-f865-11f0-8ebf-539c931c1654"`. |
| `season` | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| `competition_id` | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_player_seasons(website_id = 12, person_id = "4458f165-f865-11f0-8ebf-539c931c1654")
```

## Value

A `wehoop_data` tibble, one row per `(person_id, season_name, team_id)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| season_name | character | Competition-season the line is for, in the site's language (Betclic ELITE - Playo |
| season_year | integer | The site's own year of that season (START year on LNB). |
| season_end_year | integer | Season END year of that season (sdv key); null when the site is unmapped. |
| team_id | character | Team UUID the player played for in that season. |
| team_name | character | Name of the team the player played for in that season. |
| minutes_per_game | character | Minutes per game as an ISO 8601 duration (player, that competition-season). |
| points_per_game | double | Points per game (player, that competition-season). |
| rebounds_per_game | double | Rebounds per game (player, that competition-season). |
| rebounds_offensive_per_game | double | Offensive rebounds per game (player, that competition-season). |
| rebounds_defensive_per_game | double | Defensive rebounds per game (player, that competition-season). |
| assists_per_game | double | Assists per game (player, that competition-season). |
| steals_per_game | double | Steals per game (player, that competition-season). |
| blocks_per_game | double | Blocks per game (player, that competition-season). |
| turnovers_per_game | double | Turnovers per game (player, that competition-season). |
| fouls_personal_per_game | double | Personal fouls per game (player, that competition-season). |
| dunks_per_game | double | Dunks per game (player, that competition-season). |

| | | |
|---|---|---|
| efficiency | integer | Efficiency rating, from the site's own formula (player, that competition-season). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_stats(),
atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

# atrium_player_stats **Atrium Player Stats (season, long form)**

---

## Description

**Player season statistics in long form (one row per player, table and statistic); ids decoded from the widget links.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/statistics_persons`

Each body holds several tables (`averages`, `shootingstatistics`, `totals`, ...) whose statistic set the site configures. `stat_key` (`pointsPerGame`) carries its numeric `value` or, for ISO durations, `value_text`; pivot on `stat_key` for a wide table. Percentages are 0-100.

## Usage

```
atrium_player_stats(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see atrium_seasons()). Omit `season` and `season_id` for the site's default season. |
| season_id | (*character* optional): Season UUID (`season_id` in atrium_seasons()); sent as is, skipping the END-year lookup of `season`. |

`competition_id` (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it.

`locale` (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change.

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(table_key, person_id, stat_key)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 |
| season_id | character | Season UUID of the table; null for the competition-scoped routes (their bodies carry none). |
| table_key | character | Statistics table the row is from: averages, shootingstatistics, totals, ... (the set is per site). |
| table_label | character | The table's heading in the site's language (e.g. Moyennes). |
| person_id | character | Player UUID decoded from the widget link beside the name (the tables carry names only); null wl |
| person_name | character | Player name as the site prints it; null in team tables. |
| entity_id | character | Team UUID decoded from the widget link (team tables only; player tables name the team but carr |
| entity_name | character | Team name as the site prints it; for player rows of a stat table it is the player's team. |
| stat_key | character | Statistic key as the API names it (pointsPerGame, fieldGoalsMade, ...), unchanged from header ke |
| stat_name | character | Short column heading the site shows for the statistic (PTS, REB), in the site's language. |
| stat_label | character | Longer statistic label the site shows (code in the API), in the site's language. |
| value | double | Numeric value of the statistic; null when the API sends a duration or text (see value_text). |
| value_text | character | Non-numeric statistic value, e.g. an ISO 8601 duration such as PT12M2S for minutes per game. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

## Examples

```
try(atrium_player_stats(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
))
```

---

`atrium_players` **Atrium Players**

---

## Description

**The players of a season, with position, nationality, birth date, age and height.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/persons`

`height` is an integer where the site gives one (centimetres; unit unverified). `position_code` / `position_name` follow the site locale (the name is translated, the code is not). `person_id` is stable across seasons and sites.

## Usage

```
atrium_players(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  entity_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `season` | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |

`season_id` (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`.

`competition_id` (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it.

`entity_id` (*character* optional): Team UUID (`entity_id` in `atrium_teams()` or any fixture row).

`locale` (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change.

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(season_id, person_id)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 |
| season_id | character | Season UUID of the roster list. |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| name | character | Player name as the site prints it. |
| bib | character | Jersey number, as text. |
| entity_id | character | Team UUID the player is rostered with that season. |
| team_name | character | Team name in the site's language. |
| position_code | character | Position code (C, PG, SF, ...), language-independent. |
| position_name | character | Position name in the site's language. |

| | | |
|---|---|---|
| nationality | character | Player's nationality in the site's language (e.g. Belgique). |
| date_of_birth | Date | Player's date of birth. |
| age | integer | Player's age in years when the body was generated. |
| height | integer | Height as an integer, centimetres presumed (unit unverified); null when the site gives none. |
| status | character | Player status in the site's language (Actif, Blesse). |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

## Examples

```
try(atrium_players(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
))
```

---

`atrium_season_id` **Atrium Season Id**

---

## Description

**The `seasonId` of a site's season, from its END year and optionally its competition.**

Maps through the site's own season list (`/{website_id}/fixtures`), fetched once and cached for an hour per site.

## Usage

```
atrium_season_id(website_id, season, competition_id = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): The site (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 326 Montenegro). Site 236 lists no seasons. |
| `season` | (*integer* required): The sdv season END year (2026 = 2025-26 on LNB; the calendar year on the other sites). A whole number; a string, a logical or a fraction is refused. |
| `competition_id` | (*character* optional): The competition when the site lists several for that year (see atrium_seasons()). Omit it when the year is unique. |

## Details

Raises `wehoop_bad_request` when `season` is not a whole number, the site is not a known Atrium site, it lists no seasons (236), or the year is shared by several competitions and `competition_id` is missing; `wehoop_no_data` when the site lists no such season; `wehoop_asset_fetch` when the season list could not be fetched or is not the site's.

```
atrium_season_id(14, 2026, competition_id = "16be8396-24fb-11ef-9ab7-df6be196e88f")
```

## Value

The season's UUID (a character string), for the `season_id` argument of every season-scoped wrapper.

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_seasons()`, `atrium_shots()`, `atrium_standings()`,
`atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`,
`atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

---

# atrium_seasons Atrium Seasons

---

## Description

**Every season (competition and year) a site lists, with the sdv END year of each.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixtures`

A season is one competition in one year. `year` is the site's own key (START year on site 12, calendar year elsewhere); `season_end_year` is the sdv END year. Site 236 lists no seasons, so its frame is empty.

## Usage

```
atrium_seasons(website_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `season_id`; `is_default_season` marks the season the site serves by default:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, |
| season_id | character | Season UUID; pass it as season_id to any season-scoped wrapper. |
| competition_id | character | Competition UUID; pass it as competition_id to pick one of several seasons of an END year. |
| competition_name | character | Competition name in the site's language; the same name can appear for several years. |
| season_name | character | Season name in the site's language, e.g. Betclic ELITE 2025. |
| abbreviation_local | character | Local abbreviation of the season; null in every body seen. |
| year | integer | The site's own year of the season: the START year on LNB (site 12), the calendar year elsew |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year on th |
| is_default_season | logical | True for the season the body is about (the site's default when none was asked). |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_shots()`, `atrium_standings()`,
`atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`,
`atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

## Examples

```
try(atrium_seasons(website_id = 12))
```

---

`atrium_shots` **Atrium Shots**

---

## Description

**One game's 2pt / 3pt attempts with their court location (percent of the court) and result.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=s`

The shot chart holds field-goal attempts only (no free throws); `event_id` joins the same event in `atrium_pbp()`. `x` / `y` are percent of the court (0-100, origin at a corner, x along the 28 m length). LNBP and NBL1 clamp rim attempts to x = 5 / 95.

## Usage

```
atrium_shots(website_id, fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in `atrium_fixtures()`), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per shot (`event_id`):

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the shot among the shots of the game (not the pbp sequence). |
| event_id | character | Event UUID of the shot; the same event in atrium_pbp. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID of the shooter. |
| person_name | character | Name of the shooter. |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when the shot was made, false when missed. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_standings()`,
`atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`,
`atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

## Examples

```
try(atrium_shots(website_id = 14, fixture_id = "0dc631bd-c34c-11f1-8d16-4d5a21afb79d"))
```

`atrium_standings` **Atrium Standings**

## Description

**Standings of a season, with the columns the site configures (LNB home / away splits, LNBP played and points).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/standings`

`win_percentage` is a 0-1 fraction. `group` is the table's label (empty for a single table).

## Usage

```
atrium_standings(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  round_number = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `season` | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| `competition_id` | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |
| `round_number` | (*integer* optional): Round (journee) number to show; the pages that take it list their rounds. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the

leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(season_id, group, entity_id)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year o |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and l |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |

| | | |
|---|---|---|
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

## Examples

```
try(atrium_standings(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
))
```

---

`atrium_standings_summary`

# Atrium Standings Summary

---

## Description

**Compact standings of a season: position, wins, losses and games played.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/standings_fragment`

The columns are those of atrium_standings(); the compact fragment fills only `wins` / `losses` / `played`.

## Usage

```
atrium_standings_summary(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs competition_id when several competitions share that year (see atrium_seasons()). Omit season and season_id for the site's default season. |
| season_id | (*character* optional): Season UUID (season_id in atrium_seasons()); sent as is, skipping the END-year lookup of season. |
| competition_id | (*character* optional): Competition UUID (competition_id in atrium_seasons()). Picks the season together with season, and is then not sent; with neither season nor season_id the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. "en-EN" or "fr-FR". Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** website_id picks the league site; one site carries many competitions. Seasons are END years (season = 2026): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair season with competition_id (see atrium_seasons()) or pass season_id. atrium_end_year() maps a site's own year to that END year, and atrium_season_id() finds the season UUID.

**Columns** are configured per site: a statistic one site omits is NA. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so person_id there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); wehoop_bad_request for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, season with season_id); wehoop_asset_fetch for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_standings_summary(website_id = 12, season = 2026,
                         competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404")
```

## Value

A wehoop_data tibble, one row per (season_id, group, entity_id):

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year c |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and l |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_team()`, `atrium_team_box_score()`, `atrium_team_stats()`, `atrium_teams()`,
`lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

---

`atrium_team` **Atrium Team Page**

---

## Description

**A team page (summary tab): the team and its last and next game.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/entity_detail`

previous_* is the last played game and next_* the next scheduled one (NA when the page shows none). entity_id is decoded from the page's tab links; color is a hex string with its #.

## Usage

```
atrium_team(
  website_id,
  entity_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| entity_id | (*character* required): Team UUID (entity_id in atrium_teams() or any fixture row), e.g. "f43b1b0f-55bc-11f0-84d9-01499d2f0a07". |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs competition_id when several competitions share that year (see atrium_seasons()). Omit season and season_id for the site's default season. |
| season_id | (*character* optional): Season UUID (season_id in atrium_seasons()); sent as is, skipping the END-year lookup of season. |
| competition_id | (*character* optional): Competition UUID (competition_id in atrium_seasons()). Picks the season together with season, and is then not sent; with neither season nor season_id the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. "en-EN" or "fr-FR". Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** website_id picks the league site; one site carries many competitions. Seasons are END years (season = 2026): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair season with competition_id (see atrium_seasons()) or pass season_id. atrium_end_year() maps a site's own year to that END year, and atrium_season_id() finds the season UUID.

**Columns** are configured per site: a statistic one site omits is NA. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across

sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch` for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

```
atrium_team(website_id = 12, entity_id = "f43b1b0f-55bc-11f0-84d9-01499d2f0a07")
```

## Value

A `wehoop_data` tibble, one row for the team:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 P |
| entity_id | character | Team UUID, decoded from the page's tab links (the body does not echo it elsewhere) |
| name | character | Team name from the page banner. |
| color | character | Team colour from the banner, hex string with its #. |
| logo_url | character | URL of the banner's team logo. |
| previous_fixture_id | character | Game UUID of the last played game shown on the team page. |
| previous_round | character | Round of the last played game as the site words it (e.g. 30). |
| previous_status | character | Status code of the last played game, e.g. CONFIRMED. |
| previous_start_time_utc | POSIXct | Start of the last played game, naive UTC datetime. |
| previous_start_time_local | character | Start of the last played game in venue-local time, as the API sends it (no offset). |
| previous_venue | character | Arena of the last played game. |
| previous_attendance | integer | Spectators at the last played game; null when not reported. |
| previous_home_entity_id | character | Team UUID of the last played game's home team. |
| previous_home_name | character | Name of the last played game's home team. |
| previous_home_score | integer | Home team's points in the last played game. |
| previous_away_entity_id | character | Team UUID of the last played game's away team. |
| previous_away_name | character | Name of the last played game's away team. |
| previous_away_score | integer | Away team's points in the last played game. |
| next_fixture_id | character | Game UUID of the next scheduled game shown on the team page. |
| next_round | character | Round of the next scheduled game as the site words it (e.g. 30). |
| next_status | character | Status code of the next scheduled game, e.g. CONFIRMED. |
| next_start_time_utc | POSIXct | Start of the next scheduled game, naive UTC datetime. |
| next_start_time_local | character | Start of the next scheduled game in venue-local time, as the API sends it (no offset). |
| next_venue | character | Arena of the next scheduled game. |
| next_attendance | integer | Spectators at the next scheduled game; null when not reported. |
| next_home_entity_id | character | Team UUID of the next scheduled game's home team. |
| next_home_name | character | Name of the next scheduled game's home team. |
| next_home_score | integer | Home team's points in the next scheduled game. |
| next_away_entity_id | character | Team UUID of the next scheduled game's away team. |
| next_away_name | character | Name of the next scheduled game's away team. |
| next_away_score | integer | Away team's points in the next scheduled game. |

## See Also

Other Atrium Functions: atrium_box_score(), atrium_competition_player_stats(), atrium_competition_team_st
atrium_end_year(), atrium_fixture(), atrium_fixture_periods(), atrium_fixtures(), atrium_fixtures_ribbon
atrium_leaders(), atrium_pbp(), atrium_player(), atrium_player_game_log(), atrium_player_seasons(),
atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(),
atrium_standings(), atrium_standings_summary(), atrium_team_box_score(), atrium_team_stats(),
atrium_teams(), lmb_competition, lnbp_competition, lpb_competition, nbl1_competition

---

# atrium_team_box_score **Atrium Team Box Score**

---

## Description

**One game's team lines: the team totals, the team-level counts and the game aggregates (lead changes, biggest lead).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/fixture_detail?sub=s`

The team line is the API's `entity` block (minutes are the sum over players, `PT200M`); `extra_*` are the team-level counts not credited to a player (team rebounds and turnovers) and `total_*` the game-level aggregates.

## Usage

```
atrium_team_box_score(website_id, fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `website_id` | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| `fixture_id` | (*character* required): Game UUID (`fixture_id` in `atrium_fixtures()`), e.g. `"0dc631bd-c34c-11f1-8d16-4d5a21afb79d"`. |
| `locale` | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id`

(see atrium_seasons()) or pass season_id. atrium_end_year() maps a site's own year to that END year, and atrium_season_id() finds the season UUID.

**Columns** are configured per site: a statistic one site omits is NA. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so person_id there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); wehoop_bad_request for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, season with season_id); wehoop_asset_fetch for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A wehoop_data tibble, one row per (fixture_id, entity_id):

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 2 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail route |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID (from the fixture's isHome flag). |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome fla |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its pla |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (team, this game). |
| points | integer | Points scored (team, this game). |
| rebounds | integer | Total rebounds (team, this game). |
| rebounds_offensive | integer | Offensive rebounds (team, this game). |
| rebounds_defensive | integer | Defensive rebounds (team, this game). |
| assists | integer | Assists (team, this game). |
| steals | integer | Steals (team, this game). |
| blocks | integer | Shots blocked (team, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (team, this game). |
| turnovers | integer | Turnovers (team, this game). |
| fouls_total | integer | Fouls committed (team, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (team, this game |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lin |
| efficiency | integer | Efficiency rating, from the site's own formula (team, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (team, this game |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (team, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, this game). |
| points_two_made | integer | Two-point field goals made (team, this game). |
| points_two_attempted | integer | Two-point field goals attempted (team, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (team, this game). |
| points_three_made | integer | Three-point field goals made (team, this game). |
| points_three_attempted | integer | Three-point field goals attempted (team, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (team, this game). |

| | | |
|---|---|---|
| free_throws_made | integer | Free throws made (team, this game). |
| free_throws_attempted | integer | Free throws attempted (team, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (team, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (team, this game). |
| extra_fouls_total | integer | Fouls committed (team, credited to the team rather than a player). |
| extra_rebounds | integer | Total rebounds (team, credited to the team rather than a player). |
| extra_rebounds_defensive | integer | Defensive rebounds (team, credited to the team rather than a player). |
| extra_rebounds_offensive | integer | Offensive rebounds (team, credited to the team rather than a player). |
| extra_turnovers | integer | Turnovers (team, credited to the team rather than a player). |
| total_biggest_lead | integer | Biggest lead of the game, in points (team, game total). |
| total_biggest_scoring_run | integer | Longest unanswered scoring run of the game, in points (team, game total). |
| total_field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, game total). |
| total_free_throws_percentage | double | Free-throw percentage, 0-100 (team, game total). |
| total_lead_changes | integer | Number of lead changes in the game (team, game total). |
| total_points_fast_break | integer | Fast-break points (LNBP only) (team, game total). |
| total_points_from_bench | integer | Points scored by non-starters (LNBP only) (team, game total). |
| total_points_from_turnover | integer | Points scored off opponent turnovers (LNBP only) (team, game total). |
| total_time_in_lead | character | Time the team led, as an ISO 8601 duration (PT36M11S) (team, game total). |
| total_time_in_lead_seconds | integer | Time the team led, in whole seconds (team, game total). |
| total_turnovers | integer | Turnovers (team, game total). |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_stats()`,
`atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

## Examples

```
try(atrium_team_box_score(website_id = 12, fixture_id = "5423072b-6715-11f0-bd8b-9ffadebc148f"))
```

---

`atrium_team_stats` **Atrium Team Stats (season, long form)**

---

## Description

**Team season statistics in long form (one row per team, table and statistic).**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/statistics_entities`

## Usage

```
atrium_team_stats(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| season_id | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| competition_id | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch`

for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(table_key, entity_name, stat_key)`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 A |
| season_id | character | Season UUID of the table; null for the competition-scoped routes (their bodies carry none). |
| table_key | character | Statistics table the row is from: averages, shootingstatistics, totals, ... (the set is per site). |
| table_label | character | The table's heading in the site's language (e.g. Moyennes). |
| person_id | character | Player UUID decoded from the widget link beside the name (the tables carry names only); null wl |
| person_name | character | Player name as the site prints it; null in team tables. |
| entity_id | character | Team UUID decoded from the widget link (team tables only; player tables name the team but carr |
| entity_name | character | Team name as the site prints it; for player rows of a stat table it is the player's team. |
| stat_key | character | Statistic key as the API names it (pointsPerGame, fieldGoalsMade, ...), unchanged from header ke |
| stat_name | character | Short column heading the site shows for the statistic (PTS, REB), in the site's language. |
| stat_label | character | Longer statistic label the site shows (code in the API), in the site's language. |
| value | double | Numeric value of the statistic; null when the API sends a duration or text (see value_text). |
| value_text | character | Non-numeric statistic value, e.g. an ISO 8601 duration such as PT12M2S for minutes per game. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

## Examples

```
try(atrium_team_stats(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
))
```

---

`atrium_teams` **Atrium Teams**

---

## Description

**The teams of a season.**

Endpoint: `GET https://embed-api.eui.connect.sportradar.com/v1/embed/{website_id}/entities`

## Usage

```
atrium_teams(
  website_id,
  season = NULL,
  season_id = NULL,
  competition_id = NULL,
  locale = NULL
)
```

## Arguments

| | |
|---|---|
| website_id | (*integer* required): Site config id: which league site's widget this is (12 LNB France, 14 LNBP Mexico, 3 NBL1 Australia, 13 El Salvador, 22 Panama, 236 Austria, 326 Montenegro). Not a league id: one site serves several competitions. |
| season | (*integer* optional): Season END year (2026 = 2025-26 on LNB, site 12, whose own year is the START year; the calendar year on the other sites). Needs `competition_id` when several competitions share that year (see `atrium_seasons()`). Omit `season` and `season_id` for the site's default season. |
| season_id | (*character* optional): Season UUID (`season_id` in `atrium_seasons()`); sent as is, skipping the END-year lookup of `season`. |
| competition_id | (*character* optional): Competition UUID (`competition_id` in `atrium_seasons()`). Picks the season together with `season`, and is then not sent; with neither `season` nor `season_id` the API filters by it. |
| locale | (*character* optional): Locale of the labels, e.g. `"en-EN"` or `"fr-FR"`. Omitted = the site's own language. Ids and codes never change. |

## Details

**Unofficial.** This is the public endpoint the Sportradar Atrium / Synergy "Connect" widget calls from every visitor's browser (keyless; the host has no robots.txt). The data is Sportradar's and the leagues'. The wrappers read it on the caller's behalf, one request at least 1 s after the last to the host, and it is not republished as release assets. Keep crawls small and incremental.

**Sites and seasons.** `website_id` picks the league site; one site carries many competitions. Seasons are END years (`season = 2026`): the site's own year is the START year on LNB and the calendar year elsewhere, and a year can hold several competitions, so pair `season` with `competition_id` (see `atrium_seasons()`) or pass `season_id`. `atrium_end_year()` maps a site's own year to that END year, and `atrium_season_id()` finds the season UUID.

**Columns** are configured per site: a statistic one site omits is `NA`. A key the documented columns do not list is appended with an inferred type. Player and team ids (UUIDs, character) are shared across sites; the player stat tables carry names only, so `person_id` there is decoded from the widget link beside each name.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when the site has nothing for the request (HTTP 404, no such season, an empty fixture list, a game with no play-by-play yet, or a fixture id the site does not serve: an unknown id or a game of another site, which the API answers alike with an empty fixture); `wehoop_bad_request` for HTTP 400 / 422 or an argument that cannot be mapped (an ambiguous END year, `season` with `season_id`); `wehoop_asset_fetch`

for any other failed fetch, an error / stub body, or a body that belongs to another site, type, season, game, player or team than the one asked for.

## Value

A `wehoop_data` tibble, one row per `(season_id, entity_id)`; `entity_id` is stable across seasons:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Au |
| season_id | character | Season UUID of the team list. |
| entity_id | character | Team UUID (entity): one id per club across seasons, shared by every site that shows the team. |
| name | character | Team name in the site's language. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_team_stats()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

## Examples

```
try(atrium_teams(
 website_id = 12, season = 2026, competition_id = "3f4064bb-51ad-11f0-aaaf-2923c944b404"
))
```

---

`bart_wbb_game_schedule`

# Bart Torvik Women's Season Schedule & Results

---

## Description

**Get the full women's game-by-game schedule and results for a season from barttorvik.com.** Pulls the `/ncaaw/{year}_super_sked.json` file (one row per game). No API key required.

## Usage

```
bart_wbb_game_schedule(year = most_recent_wbb_season())
```

## Arguments

| | |
|---|---|
| `year` | Season, 4-digit ending-year format (e.g. `2024`). Defaults to `most_recent_wbb_season()`. |

## Value

A wehoop_data tibble, one row per game: muid, date, team1, team2, t1pts, t2pts, winner, loser, plus efficiency / quality columns, and year.

## See Also

Other Bart Torvik Functions: bart_wbb_ratings()

## Examples

```
try(bart_wbb_game_schedule(year = 2024))
```

---

# bart_wbb_ratings Bart Torvik Women's T-Rank Ratings

---

## Description

**Get women's college basketball T-Rank team ratings and adjusted efficiencies from bart-torvik.com.** Pulls the /ncaaw/{year}_team_results.csv file (one row per team). No API key required.

## Usage

```
bart_wbb_ratings(year = most_recent_wbb_season())
```

## Arguments

| | |
|---|---|
| year | Season, 4-digit ending-year format (e.g. 2024). Defaults to most_recent_wbb_season(). |

## Value

A wehoop_data tibble, one row per team: rank, team, conf, record, adjoe, adjde, barthag, plus projection / SOS / quality columns, and year.

## See Also

Other Bart Torvik Functions: bart_wbb_game_schedule()

## Examples

```
try(bart_wbb_ratings(year = 2024))
```

bref_wnba_player_stats

## Basketball-Reference WNBA Player Season Stats

### Description

**Get WNBA league-wide player season statistics from Basketball-Reference.**

One row per player for a season. No account or API key is required. Basketball-Reference rate-limits aggressive scraping (~20 requests/minute) – space repeated calls with `Sys.sleep()`.

### Usage

```
bref_wnba_player_stats(season = most_recent_wnba_season(), table = "per_game")
```

### Arguments

| | |
|---|---|
| season | Season, in 4-digit year format (e.g. 2024). Defaults to `most_recent_wnba_season()`. |
| table | Which stats table – one of `"per_game"` (default), `"totals"` or `"advanced"`. |

### Value

A wehoop_data tibble with one row per player, carrying season and table columns.

| col_name | types |
|---|---|
| player | character |
| team_id | character |
| pos | character |
| g | numeric |
| pts_per_g | numeric |
| season | integer |
| table | character |

### See Also

Other WNBA Basketball-Reference Functions: `bref_wnba_standings()`, `bref_wnba_team_stats()`

### Examples

```
try(bref_wnba_player_stats(season = 2024, table = "per_game"))
```

`bref_wnba_standings` **Basketball-Reference WNBA Standings**

## Description

**Get WNBA standings (both conferences) from Basketball-Reference.**

One row per team with record and win percentage, labelled by a `conference` column (`Eastern` / `Western`). No account or API key is required.

## Usage

```
bref_wnba_standings(season = most_recent_wnba_season())
```

## Arguments

| | |
|---|---|
| `season` | Season, 4-digit year format (e.g. `2024`). Defaults to `most_recent_wnba_season()`. |

## Value

A `wehoop_data` tibble with one row per team:

| col_name | types |
|---|---|
| team | character |
| wins | numeric |
| losses | numeric |
| win_loss_pct | numeric |
| conference | character |
| season | integer |

## See Also

Other WNBA Basketball-Reference Functions: `bref_wnba_player_stats()`, `bref_wnba_team_stats()`

## Examples

```
try(bref_wnba_standings(season = 2024))
```

`bref_wnba_team_stats` **Basketball-Reference WNBA Team Season Stats**

## Description

**Get WNBA team season statistics from Basketball-Reference.**

One row per team for a season. No account or API key is required.

## Usage

```
bref_wnba_team_stats(season = most_recent_wnba_season(), table = "per_game")
```

## Arguments

| | |
|---|---|
| `season` | Season, 4-digit year format (e.g. `2024`). Defaults to `most_recent_wnba_season()`. |
| `table` | Which stats table – one of `"per_game"` (default), `"totals"`, `"per_poss"` or `"advanced"`. |

## Value

A `wehoop_data` tibble with one row per team, carrying `season` and `table` columns.

| col_name | types |
|---|---|
| team | character |
| g | numeric |
| pts_per_g | numeric |
| season | integer |
| table | character |

## See Also

Other WNBA Basketball-Reference Functions: `bref_wnba_player_stats()`, `bref_wnba_standings()`

## Examples

```
try(bref_wnba_team_stats(season = 2024, table = "per_game"))
```

## check_status *Check Status function*

### Description

Check Status function

### Usage

```
check_status(res)
```

### Arguments

| | |
|---|---|
| res | Response from an httr2 request |

## csv_from_url **Load .csv / .csv.gz file from a remote connection**

### Description

This is a thin wrapper on data.table::fread

### Usage

```
csv_from_url(...)
```

### Arguments

| | |
|---|---|
| ... | passed to data.table::fread |

## espn_basketball **ESPN Basketball Endpoint Overview (WNBA + WBB)**

### Description

Wrappers around ESPN's basketball endpoints. Two parallel families share a common set of internal helpers in `R/espn_basketball_*.R`:

- `espn_wnba_*()` — ESPN WNBA wrappers
- `espn_wbb_*()` — ESPN women's college basketball wrappers

Each public wrapper is a thin shim over an internal helper that takes a `league` argument (`"wnba"` or `"womens-college-basketball"`); the helper does the actual HTTP call + parsing. Error reporting flows through ESPN-side reporters `.report_api_error()` / `.report_api_warning()` in `R/utils.R` (do **not** use raw `cli::cli_alert_danger()` here — that is the WNBA Stats convention).

## Details

### Play-by-play, scoreboard, schedule:

| WNBA function | WBB function |
|---|---|
| espn_wnba_pbp() | espn_wbb_pbp() |
| espn_wnba_scoreboard() | espn_wbb_scoreboard() |
| espn_wnba_game_all() | espn_wbb_game_all() |
| espn_wnba_game_rosters() | espn_wbb_game_rosters() |
| espn_wnba_player_box() | espn_wbb_player_box() |
| espn_wnba_team_box() | espn_wbb_team_box() |

### Reference data:

| WNBA function | WBB function |
|---|---|
| espn_wnba_teams() | espn_wbb_teams() |
| espn_wnba_standings() | espn_wbb_standings() |
| espn_wnba_player_stats() | espn_wbb_player_stats() |
| espn_wnba_team_stats() | espn_wbb_team_stats() |
| | espn_wbb_conferences() |
| | espn_wbb_rankings() |

### HTTP layer:

ESPN wrappers call `.retry_request()` directly without `...`, so per-call proxy overrides aren't supported. Use `options(wehoop.proxy = ...)` or the `http_proxy` / `https_proxy` env vars for proxy routing.

## See Also

Other ESPN Basketball: espn_basketball_player_core()

---

`espn_basketball_player_core` **Project an ESPN core-v2 athlete record into a `player_core` row**

---

## Description

Turns one ESPN core-v2 `/athletes/{id}` payload into the single tidy row released as the `player_core` dataset.

This is a **pure projection**: it takes an already-fetched payload and never performs I/O. That is deliberate — the compile stages in `wehoop-wnba-data` / `wehoop-wbb-data` read the payload from the sibling `-raw` tree, and re-fetching here would both break the one-way raw -> data boundary and mean the R and Python pipelines read different bytes, so a value divergence could not be attributed to method.

Ids for college and current team are parsed out of the payload's `$ref` URLs (`/colleges/{id}`, `/teams/{id}`). The `$ref` is **never** followed.

## Usage

```
espn_basketball_player_core(payload, athlete_id)
```

## Arguments

| | |
|---|---|
| payload | list. One athlete's core-v2 `/athletes/{athlete_id}` payload, as returned by `jsonlite::fromJSON(..., simplifyVector = FALSE)`. An empty or non-list value yields a zero-row tibble rather than an error. |
| athlete_id | numeric or character. The ESPN athlete id. **Required and never inferred from the payload** — callers pass the id from the file path, so a payload missing its own `id` still produces a joinable row. |

## Details

**What the row means.** `current_team_id` is the athlete's team *today*, not their team in any past season — the season a released row is filed under is *participation* (who appeared that year, taken from `player_box`), not the vintage of the bio. Height, weight and jersey are likewise a current snapshot: ESPN overwrites them in place, so era-correct bio is not obtainable from this endpoint.

**Parity.** This is a port of `sportsdataverse.wbb.helper_wbb_player_core` (sdv-py 0.0.75), which produces the released dataset today. The two are held to byte-parity by `tests/testthat/test-espn_basketball_player_c` against a golden fixture captured from that function; see `tests/testthat/fixtures/player_core/README.md` for provenance. Neither implementation is authoritative — a divergence is a review item.

## Value

A one-row tibble carrying the full 35-column set (absent fields are `NA`), so callers see a stable schema regardless of payload completeness:

See the HTML help or pkgdown reference for the column table.

## Twin

`hoopR::espn_basketball_player_core()` is the identical function for the men's leagues. The core-v2 athlete resource is the same payload shape for nba/wnba/mbb/wbb, so the projection is league-agnostic – sdv-py implements it once and re-exports it per league. hoopR and wehoop are independently published and neither depends on the other, so here it is duplicated: **a change to one must land in the other in the same session, verified.**

## Author(s)

Saiem Gilani

## See Also

Other ESPN Basketball: `espn_basketball`

## Examples

```
# Split across lines to keep the Rd under the line-width limit; the
# core-v2 $ref URLs are long enough to be truncated in the PDF manual.
team_ref <- paste0(
  "http://sports.core.api.espn.com/v2/sports/basketball/",
  "leagues/wnba/seasons/2025/teams/22"
)
payload <- list(
  guid = "abc", fullName = "Jane Doe", jersey = "23",
  position = list(id = "5", abbreviation = "G"),
  team = list(`$ref` = team_ref)
)
espn_basketball_player_core(payload, athlete_id = 1966)
```

---

espn_wbb_athletes_index

# Get ESPN Women's College Basketball Athletes Index

---

## Description

**Get ESPN Women's College Basketball Athletes Index**

**Get ESPN Women's College Basketball Athletes Index**

## Usage

```
espn_wbb_athletes_index(
  season = most_recent_wbb_season(),
  active = TRUE,
  limit = 25000L,
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric, e.g. 2025). Defaults to the most recent WBB season. |
| active | logical. When TRUE (default) only active athletes are returned. Set to FALSE for the full historical roster. |
| limit | integer. Maximum number of rows to return. Default 25000. WBB can return 6,000-12,000 athletes per season; increase if needed. Pass a small value (e.g. limit = 50) in tests to keep execution fast. |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use options(wehoop.proxy = ...) – see ?wehoop for details. |

## Value

A single wehoop_data tibble with one row per athlete.

Columns as documented in the shared espn_basketball_athletes_index_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_athletes_index(season = 2025, limit = 50)
```

---

# espn_wbb_award — Get ESPN WBB Season Award Detail

---

## Description

Returns the name, description, and winners of one WBB season award.

## Usage

```
espn_wbb_award(award_id, season = most_recent_wbb_season(), ...)
```

### Arguments

| | |
|---|---|
| award_id | ESPN award identifier (character or numeric). |
| season | Season year (numeric). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused. |

### Value

A tibble with one row per winner.

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

### Examples

```
espn_wbb_award(award_id = 344, season = 2024)
```

---

## espn_wbb_calendar Get ESPN WBB Calendar

---

### Description

**Get ESPN WBB Calendar**

**Get ESPN Women's College Basketball Calendar**

## Usage

```
espn_wbb_calendar(season = most_recent_wbb_season())
```

## Arguments

| | |
|---|---|
| season | integer or character. Four-digit season year (e.g. 2025). Defaults to most_recent_wbb_season(). |

## Details

Retrieve the ESPN women's college basketball schedule calendar for a given season. The underlying scoreboard response contains a leagues[[1]]$calendar block with season-type entries (pre-season, regular, post). Uses getOption("wehoop.proxy") or http_proxy/https_proxy environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of calendar entries.

Columns as documented in the shared espn_basketball_calendar_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_calendar(season = 2025)
```

`espn_wbb_coach` **Get ESPN WBB Single-Coach Detail**

## Description

Returns biography, current team / college refs, and counts of career record entries + per-season coaching entries for one WBB head coach.

## Usage

```
espn_wbb_coach(coach_id, ...)
```

## Arguments

| | |
|---|---|
| `coach_id` | ESPN coach identifier (character or numeric). |
| `...` | Additional arguments; currently unused. |

## Value

A single-row tibble.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personnel()`, `espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings()`, `espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`, `espn_wbb_season_weeks()`, `espn_wbb_seasons()`, `espn_wbb_standings()`, `espn_wbb_team()`, `espn_wbb_team_injuries()`, `espn_wbb_team_news()`, `espn_wbb_team_record_detail()`, `espn_wbb_team_schedule()`, `espn_wbb_team_season_roster()`, `espn_wbb_team_season_statistics()`, `espn_wbb_team_stats()`, `espn_wbb_teams()`, `espn_wbb_tournament()`, `espn_wbb_tournament_season()`, `espn_wbb_tournament_seasons()`, `espn_wbb_tournaments()`, `espn_wbb_venues()`, `espn_wbb_week_ranking()`, `espn_wbb_week_rankings()`

## Examples

```
espn_wbb_coach(coach_id = 2167842)
```

---

`espn_wbb_coach_record` **Get ESPN WBB Coach Career Record (Long Format)**

---

## Description

Returns a coach's career record by type in long format (one row per stat in the record's `stats[]` array). `record_type` codes commonly populated: 0 = Total, 1 = Pre Season, 2 = Regular Season, 3 = Post Season. Use `espn_wbb_coaches()` to discover coach_ids for a season.

## Usage

```
espn_wbb_coach_record(coach_id, record_type = 0L, ...)
```

## Arguments

| | |
|---|---|
| `coach_id` | ESPN coach identifier (use `espn_wbb_coaches()` to find). |
| `record_type` | Integer record type: 0 = Total (default), 1 = Pre Season, 2 = Regular Season, 3 = Post Season. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personnel()`, `espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings()`, `espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`, `espn_wbb_season_weeks()`, `espn_wbb_seasons()`, `espn_wbb_standings()`,

espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_coach_record(coach_id = 2167842, record_type = 2)
```

---

# espn_wbb_coach_season Get ESPN WBB Coach-in-Season Detail

---

## Description

Per-season coach detail (name, birth info, $refs to team/college/ person). ESPN's coverage of this endpoint is sparse; many (coach × season) combinations return 404.

## Usage

```
espn_wbb_coach_season(coach_id, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| coach_id | ESPN coach identifier. |
| season | Season year. Defaults to most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Coach-in-Season Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_coach_season_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_coach_season(coach_id = 2167842, season = 2025)
```

---

`espn_wbb_coaches` **Get ESPN Women's College Basketball Coaches**

---

## Description

**Get ESPN Women's College Basketball Coaches**

**Get ESPN Women's College Basketball Coaches**

## Usage

```
espn_wbb_coaches(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric, e.g. 2025). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single wehoop_data tibble with one row per coach.

| col_name | types | description |
|---|---|---|
| coach_id | character | Unique identifier for coach. |
| first_name | character | Player's first name. |
| last_name | character | Player's last name. |
| full_name | character | Player's full name. |
| experience | integer | Years of professional experience. |
| team_id | character | Unique team identifier. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_coaches(season = 2025)
```

---

# espn_wbb_conferences *Get women's college basketball conferences*

---

## Description

Get women's college basketball conferences

## Usage

```
espn_wbb_conferences()
```

## Value

Returns a tibble

Columns as documented in the shared espn_basketball_conferences_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_conferences())
```

---

# espn_wbb_franchise Get ESPN WBB Franchise Detail

---

## Description

Returns franchise-level metadata for a women's college basketball program. Franchise IDs are stable across program reorganization or rebrands.

## Usage

```
espn_wbb_franchise(franchise_id, ...)
```

## Arguments

| | |
|---|---|
| franchise_id | ESPN franchise identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_franchise(franchise_id = 2509)
```

`espn_wbb_franchises` **Get ESPN WBB Franchises Index**

## Description

Returns the full WBB franchises index from `sports.core.api.espn.com/v2/sports/basketball/leagues/womens-col`
Each row is one franchise with its ID and the canonical `$ref` URL — pass an ID to `espn_wbb_franchise()`
for full franchise detail.

## Usage

```
espn_wbb_franchises(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per franchise.

| col_name | types | description |
|---|---|---|
| franchise_id | character | ESPN franchise identifier. |
| ref | character | Full `$ref` URL for franchise detail. |
| league | character | League slug (`"womens-college-basketball"`). |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personnel()`, `espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings()`, `espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`, `espn_wbb_season_weeks()`, `espn_wbb_seasons()`, `espn_wbb_standings()`, `espn_wbb_team()`, `espn_wbb_team_injuries()`, `espn_wbb_team_news()`, `espn_wbb_team_record_detail()`,

espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_franchises()
```

---

`espn_wbb_game_all` *Get ESPN women's college basketball game data (play-by-play, team and player box)*

---

## Description

Get ESPN women's college basketball game data (play-by-play, team and player box)

Get ESPN women's college basketball play by play data

Get ESPN women's college basketball team box data

Get ESPN women's college basketball player box

**Get ESPN women's college basketball game rosters**

## Usage

```
espn_wbb_game_all(game_id)

espn_wbb_pbp(game_id)

espn_wbb_team_box(game_id)

espn_wbb_player_box(game_id)

espn_wbb_game_rosters(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | Game ID |

## Value

A named list of dataframes: Plays, Team, Player

**Plays**

Columns as documented in the shared espn_basketball_game_all_plays_plays_schema table.

**Team**

Columns as documented in the shared espn_basketball_team_box_schema table.

**Player**

Columns as documented in the shared espn_basketball_game_all_player_player_schema table.

Returns a play-by-play data frame

**Plays**

Columns as documented in the shared espn_basketball_game_all_plays_plays_schema table.

Returns a team boxscore data frame

**Team**

Columns as documented in the shared espn_basketball_team_box_schema table.

Returns a player boxscore data frame

**Player**

Columns as documented in the shared espn_basketball_game_all_player_player_schema table.

A game rosters data frame

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_game_all(game_id = 401276115))


try(espn_wbb_pbp(game_id = 401498717))
```

```
try(espn_wbb_team_box(game_id = 401276115))

try(espn_wbb_player_box(game_id = 401276115))

try(espn_wbb_game_rosters(game_id = 401276115))
```

---

`espn_wbb_game_odds` **Get ESPN Women's College Basketball Event Odds**

---

## Description

Returns the live game situation for one WBB event: timeouts remaining, team fouls, fouls to give, bonus state, and a `$ref` to the last play. During a live game this reflects current state; after the game ends the values are frozen.

Returns pre-game predictor statistics for one WBB event in long format: one row per (team × statistic). Typical stats include matchup quality, predicted score, win probability, and team strength metrics. Returns empty for events without predictor data (often the case for already-played games).

Returns the per-team power-index `$ref` URLs for one WBB event. Coverage is sparse — many events return zero items.

## Usage

```
espn_wbb_game_odds(event_id, ...)

espn_wbb_game_officials(event_id, ...)

espn_wbb_game_broadcasts(event_id, ...)

espn_wbb_game_situation(event_id, ...)

espn_wbb_game_predictor(event_id, ...)

espn_wbb_game_powerindex(event_id, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Event Predictor (Pre-game)**

**Get ESPN WBB Event Power Index Index**

## Value

A tibble with one row per odds provider (typically empty for WBB because ESPN does not carry NCAA basketball betting lines).

Columns as documented in the shared espn_basketball_game_odds_schema table.

A tibble with one row per official assigned to the game.

Columns as documented in the shared espn_basketball_game_officials_schema table.

A tibble with one row per broadcast outlet for the game.

Columns as documented in the shared espn_basketball_game_broadcasts_schema table.

A single-row tibble with timeouts + fouls for both teams.

A long tibble with rows for both home and away teams.

A tibble with one row per team-game power-index entry.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_odds(event_id = "401276115")


espn_wbb_game_officials(event_id = "401276115")


espn_wbb_game_broadcasts(event_id = "401276115")
```

```
espn_wbb_game_situation(event_id = 401276115)

espn_wbb_game_predictor(event_id = 401276115)

espn_wbb_game_powerindex(event_id = 401276115)
```

---

`espn_wbb_game_official_detail`

**Get ESPN WBB Event Official Detail (Single Official)**

---

## Description

Returns a single-row tibble for one referee assigned to one WBB event, with their name, position (Referee / Crew Chief / Umpire), and crew order. Pair with `espn_wbb_game_officials()` to enumerate the crew.

## Usage

```
espn_wbb_game_official_detail(event_id, order, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `order` | Crew order index (1 = first official). Pair with the `order` column from event_officials(). |
| `...` | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personnel()`, `espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`,

espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_official_detail(event_id = 401276115, order = 1)
```

---

# espn_wbb_game_play Get ESPN WBB Event Play Detail (Single Play)

---

## Description

Returns the rich detail block for a single WBB play: sequence, period, clock, text, scoring/shooting flags, current home/away score, team $ref, and shot coordinates if applicable. Complements the bulk espn_wbb_pbp() output by exposing the canonical core-v2 play record.

## Usage

```
espn_wbb_game_play(event_id, play_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| play_id | ESPN play identifier (visible in pbp() output as play_id or extractable from plays[].$ref). |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play_personnel(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_play(event_id = 401276115, play_id = 401276115101899901)
```

---

`espn_wbb_game_play_personnel`

# Get ESPN WBB Event Play Personnel (On-Court Lineup at Play)

---

## Description

Returns the players on court at a specific WBB play in long format (one row per athlete entry across both competitors). Foundation for lineup analysis. ESPN coverage is sparse — many plays return zero rows; the wrapper returns a typed empty tibble in that case.

## Usage

```
espn_wbb_game_play_personnel(event_id, play_id, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `play_id` | ESPN play identifier. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble with one row per on-court athlete.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros, espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_play_personnel(event_id = 401276115, play_id = 401276115101899901)
```

---

`espn_wbb_game_player_box`

# Get ESPN WBB Event Player Box Score (Long Format)

---

## Description

Returns the long-format per-game box score for a single athlete in one WBB event. One row per (category x stat). Same shape as espn_wbb_game_team_statistics() but scoped to a single athlete-in-event instead of the full team. `stat_type` defaults to 0 (regular-season aggregate as ESPN tags it for finished events).

## Usage

```
espn_wbb_game_player_box(event_id, team_id, athlete_id, stat_type = 0L, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| team_id | ESPN team identifier (the competitor the athlete played for). |
| athlete_id | ESPN athlete identifier. |
| stat_type | Integer stat-type segment. Defaults to 0 (the only type commonly populated for finished events). |
| ... | Additional arguments; currently unused. |

## Value

A long tibble with one row per (category x stat).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_player_box(event_id = 401276115, team_id = 52,
                         athlete_id = 4433404)
```

espn_wbb_game_probabilities

**Get ESPN Women's College Basketball Event Win Probabilities**

## Description

**Get ESPN Women's College Basketball Event Win Probabilities**

**Get ESPN Women's College Basketball Event Win Probabilities**

## Usage

```
espn_wbb_game_probabilities(event_id, limit = 200, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event/game identifier (character or numeric). |
| limit | integer. Maximum number of probability rows to return. Defaults to 200. Pagination is handled internally. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per play-level win-probability entry.

Columns as documented in the shared espn_basketball_game_probabilities_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(),

espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_probabilities(event_id = "401276115", limit = 50)
```

---

`espn_wbb_game_propbets`

# Get ESPN WBB Event Prop Bets (Long Format)

---

## Description

Returns prop-bet markets for one WBB event + provider in long format: one row per (athlete × prop type). Each row has american / decimal / fraction odds plus the current target (e.g. over/under line). Hits the core-v2 `competitions/{id}/odds/{provider_id}/propBets` endpoint and auto-paginates.

## Usage

```
espn_wbb_game_propbets(event_id, provider_id, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `provider_id` | Sportsbook provider id (e.g. 58 = ESPN BET, 100 = Caesars). Look up via espn_wbb_game_odds(). |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Event Prop Bets (Long Format)**

## Value

A long tibble with one row per (athlete × prop type).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_propbets(event_id = 401276115, provider_id = 58)
```

---

`espn_wbb_game_team_linescores`

# Get ESPN WBB Event Competitor Linescores (Per-Quarter)

---

## Description

Returns the per-quarter scoring breakdown for one team in one NBA event. One row per period (regulation quarters + any overtime periods).

Returns the per-team statistical leaders for one WBB event in long format: one row per (category x athlete rank). Categories typically include points, rebounds, assists, and rating.

Returns the game-day roster index for one team in one WBB event. Each row carries the athlete id and the core-v2 `$ref` URL — use the ref to dereference athlete-game splits or biographical data.

Returns full team-game statistics for one team in one WBB event in long format: one row per (category x stat). Covers offensive, defensive, and general categories with both raw values and display strings.

Returns team records as of the given WBB event: overall, home, away, conference, and division breakdowns where available. One row per record type.

Returns a one-row tibble with one team's final score for one WBB event: numeric `value`, display string, `winner` flag, and source metadata. Quick-lookup wrapper — use espn_wbb_game_team_linescores() for per-period detail.

## Usage

```
espn_wbb_game_team_linescores(event_id, team_id, ...)

espn_wbb_game_team_leaders(event_id, team_id, ...)

espn_wbb_game_team_roster(event_id, team_id, ...)

espn_wbb_game_team_statistics(event_id, team_id, ...)

espn_wbb_game_team_records(event_id, team_id, ...)

espn_wbb_game_team_score(event_id, team_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| team_id | ESPN team identifier. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per period.

A long tibble with one row per (category x rank).

A tibble with one row per active athlete.

A long tibble with one row per (category x stat).

A tibble with one row per record type.

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(),

espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros
espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(),
espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(),
espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_team_linescores(event_id = 401276115, team_id = 52)

espn_wbb_game_team_leaders(event_id = 401276115, team_id = 52)

espn_wbb_game_team_roster(event_id = 401276115, team_id = 52)

espn_wbb_game_team_statistics(event_id = 401276115, team_id = 52)

espn_wbb_game_team_records(event_id = 401276115, team_id = 52)

espn_wbb_game_team_score(event_id = 401276115, team_id = 52)
```

---

# espn_wbb_game_team_roster_entry

**Get ESPN WBB Event Competitor Roster Entry (Per-Athlete Game-Day Row)**

---

## Description

Returns a single-row tibble describing one athlete's game-day roster entry for one WBB event. Carries the **starter** flag, **didNotPlay** flag with reason, ejection flag, and the substitution slot if the athlete came in for another player. Pair with espn_wbb_game_team_roster() to enumerate the roster.

## Usage

```
espn_wbb_game_team_roster_entry(event_id, team_id, athlete_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| team_id | ESPN team identifier. |
| athlete_id | ESPN athlete identifier. |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_game_team_roster_entry(event_id = 401276115,
                                          team_id = 2509,
                                          athlete_id = 4433404)
```

---

# espn_wbb_injuries Get ESPN Women's College Basketball Injuries

---

## Description

**Get ESPN Women's College Basketball Injuries**

**Get ESPN Women's College Basketball Injuries**

## Usage

```
espn_wbb_injuries(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Numeric or character season year (e.g. 2025). The ESPN injury endpoint does not filter by season server-side; the value is attached as a constant column on the returned tibble for downstream joins. Defaults to most_recent_wbb_season(). |
| ... | Currently unused; reserved for future argument threading. |

## Value

Returns a tibble of league-wide WBB injury records. ESPN's WBB injury data is typically sparse; the function returns an empty tibble (zero rows) when no injuries are reported.

### Injuries

Columns as documented in the shared espn_basketball_injuries_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2( espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_injuries()
```

`espn_wbb_leaders` **Get ESPN Women's College Basketball League Leaders**

## Description

**Get ESPN Women's College Basketball League Leaders**

**Get ESPN Women's College Basketball League Leaders**

## Usage

```
espn_wbb_leaders(season = most_recent_wbb_season(), season_type = 2, ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WBB season. |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single `wehoop_data` tibble with one row per category-athlete pair.

Columns as documented in the shared espn_basketball_leaders_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2(`
`espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings(`
`espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`, `espn_wbb_season_weeks()`, `espn_wbb_seasons()`, `espn_wbb_standings()`, `espn_wbb_team()`, `espn_wbb_team_injuries()`, `espn_wbb_team_news()`, `espn_wbb_team_record_detail()`, `espn_wbb_team_schedule()`, `espn_wbb_team_season_ros`
`espn_wbb_team_season_statistics()`, `espn_wbb_team_stats()`, `espn_wbb_teams()`, `espn_wbb_tournament()`, `espn_wbb_tournament_season()`, `espn_wbb_tournament_seasons()`, `espn_wbb_tournaments()`, `espn_wbb_venues()`, `espn_wbb_week_ranking()`, `espn_wbb_week_rankings()`

## Examples

```
espn_wbb_leaders(season = 2025, season_type = 2)
```

---

espn_wbb_news **Get ESPN WBB News**

---

## Description

**Get ESPN WBB News**

**Get ESPN Women's College Basketball News**

## Usage

```
espn_wbb_news(limit = 50)
```

## Arguments

| | |
|---|---|
| limit | integer. Maximum number of articles to return. Default 50. |

## Details

Retrieve ESPN women's college basketball news. Uses `getOption("wehoop.proxy")` or `http_proxy`/`https_proxy` environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of news articles.

Columns as documented in the shared espn_basketball_news_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_ espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(),

espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(
espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(),
espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(),
espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros
espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(),
espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(),
espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_news(limit = 5)
```

---

espn_wbb_player_awards

# Get ESPN Women's College Basketball Athlete Awards

---

## Description

**Get ESPN Women's College Basketball Athlete Awards**

**Get ESPN Women's College Basketball Athlete Awards**

## Usage

```
espn_wbb_player_awards(athlete_id, ...)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

## Value

A single tibble. This endpoint is sparse; many athletes have no award data, in which case an empty tibble with canonical columns is returned.

Columns as documented in the shared espn_basketball_player_awards_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_player_awards(athlete_id = "4433404")
```

---

`espn_wbb_player_career_stats`

# Get ESPN WBB Athlete Career Stats (Long Format)

---

## Description

Returns career stats for a WBB athlete in long format. Default `stat_type = 0L` fetches the standard "All Splits" / regular-season view. Pass a vector like `c(0L, 1L, 2L)` to attempt multiple types and bind them via a `stat_type_id` column; variants that 404 for that athlete are silently skipped. Stat types: 0 = regular season (default endpoint), 1 = postseason, 2 = career aggregate. Coverage of types 1 and 2 is sparse — many athletes only have type 0 populated.

## Usage

```
espn_wbb_player_career_stats(athlete_id, stat_type = 0L, ...)
```

## Arguments

| | |
|---|---|
| `athlete_id` | ESPN athlete identifier. |
| `stat_type` | Integer or integer vector of stat-type codes. Default `0L` fetches the standard "All Splits" / regular-season view. Pass a vector like `c(0L, 1L, 2L)` to bind multiple types via a `stat_type_id` column; non-existent variants are silently skipped. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Athlete Career Stats (Long Format)**

## Value

A long tibble (one row per stat_type × split × category × stat).

Columns as documented in the shared espn_basketball_player_career_stats_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
# LeBron James — regular + postseason combined
espn_wbb_player_career_stats(athlete_id = 4433404)
# Just career aggregate
espn_wbb_player_career_stats(athlete_id = 4433404, stat_type = 2L)
```

---

`espn_wbb_player_eventlog_v2`

**Get ESPN WBB Athlete Per-Season Event Log (core-v2)**

---

## Description

Returns one row per (event x team) for an WBB athlete's appearances in a given season. Distinct from espn_wbb_player_eventlog() which wraps the web-common-v3 endpoint; this core-v2 variant is era-correct.

## Usage

```
espn_wbb_player_eventlog_v2(athlete_id, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier. |
| season | Season year. Defaults to most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per event appearance.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings( espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_ros espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_player_eventlog_v2(athlete_id = 4433404, season = 2025)
```

`espn_wbb_player_info` **Get ESPN Women's College Basketball Athlete Info**

## Description

**Get ESPN Women's College Basketball Athlete Info**

**Get ESPN Women's College Basketball Athlete Info**

## Usage

```
espn_wbb_player_info(athlete_id, ...)
```

## Arguments

| | |
|---|---|
| `athlete_id` | ESPN athlete identifier (character or numeric). |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A named list of data frames: `Bio`, `Team`, `Position`, `Status`, `College`, `Draft`.

**Bio**

Columns as documented in the shared espn_basketball_player_info_bio_schema table.

**Team**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| abbreviation | character | Short abbreviation. |
| display_name | character | Display name. |

**Position**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |
| abbreviation | character | Short abbreviation. |

**Status**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |

| | | |
|---|---|---|
| type | character | Record type / category. |

**College**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |
| mascot | character | Team mascot. |

**Draft**

| col_name | types | description |
|---|---|---|
| year | character | 4-digit year. |
| round | character | Tournament / playoff round. |
| selection | character | Selection. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_player_info(athlete_id = "4433404")
```

espn_wbb_player_overview

# Get ESPN Women's College Basketball Athlete Overview

## Description

**Get ESPN Women's College Basketball Athlete Overview**

**Get ESPN Women's College Basketball Athlete Stats**

**Get ESPN Women's College Basketball Athlete Gamelog**

**Get ESPN Women's College Basketball Athlete Splits**

**Get ESPN Women's College Basketball Athlete Eventlog**

**Get ESPN Women's College Basketball Athlete Eventlog**

**Get ESPN Women's College Basketball Athlete Statisticslog**

## Usage

```
espn_wbb_player_overview(athlete_id, season = most_recent_wbb_season(), ...)

espn_wbb_player_stats_v3(athlete_id, season = most_recent_wbb_season(), ...)

espn_wbb_player_gamelog(athlete_id, season = most_recent_wbb_season(), ...)

espn_wbb_player_splits(athlete_id, season = most_recent_wbb_season(), ...)

espn_wbb_player_eventlog(athlete_id, season = most_recent_wbb_season(), ...)

espn_wbb_player_statisticslog(
  athlete_id,
  season = most_recent_wbb_season(),
  ...
)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| season | Season year (numeric). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Value

A named list of data frames: Statistics, NextGame, Last5Games, Headlines, FantasyOutlook.

**Statistics**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

**NextGame**

Columns as documented in the shared espn_basketball_player_overview_next_game_schema table.

**Last5Games**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

**Headlines**

| col_name | types | description |
|---|---|---|
| headline | character | News headline. |
| description | character | Long-form description text. |
| published | character | Publication timestamp (ISO 8601). |

**FantasyOutlook**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

A wide `wehoop_data` tibble, one row per athlete-season-team, with the ESPN stat categories spread across prefixed columns:

| col_name | types | description |
|---|---|---|
| athlete_id | character | ESPN athlete identifier (echoed input). |
| season | integer | Season year for the stat line. |
| team_id | character | ESPN team identifier for that season. |
| team_slug | character | Team slug (e.g. 'iowa-hawkeyes'). |
| avg_* | numeric | Per-game season-average stats (e.g. `avg_avg_points`). |
| tot_* | numeric | Season-total stats (e.g. `tot_points`). |
| misc_* | numeric | Miscellaneous season totals. |

Stat column names come from ESPN's positional `names` array per category, cleaned via `janitor::make_clean_names()`; the exact set varies by league and season.

A single tibble with one row per game. Column names reflect the stat labels returned by ESPN and will vary by season and player.

A single long-format tibble. When data are present, columns include at minimum `category` and `split_name`, plus per-stat columns driven by ESPN labels.

A single tibble. Per-event `statistics.$ref` URLs from the ESPN core-v2 API are returned as the character column `statistics_ref` and are NOT resolved. Similarly, `event_ref`, `competition_ref`, and `team_ref` are returned as character columns.

| col_name | types | description |
|---|---|---|
| event_ref | character | Reference link to the originating event. |
| competition_ref | character | Competition ref. |
| team_ref | character | Team ref. |
| statistics_ref | character | Statistics ref. |

A single tibble. When resolved, each row corresponds to one statistical entry in the core-v2 statistics log, with `event_ref` and `statistics_ref` character columns pointing to resolvable ESPN endpoints.

| col_name | types | description |
|---|---|---|
| event_ref | character | Reference link to the originating event. |
| statistics_ref | character | Statistics ref. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_player_overview(athlete_id = "4433404", season = 2025)

espn_wbb_player_stats_v3(athlete_id = "4433985", season = 2025)

espn_wbb_player_gamelog(athlete_id = "4433404", season = 2025)

espn_wbb_player_splits(athlete_id = "4433404", season = 2025)

espn_wbb_player_eventlog(athlete_id = "4433404", season = 2025)

espn_wbb_player_statisticslog(athlete_id = "4433404", season = 2025)
```

`espn_wbb_player_seasons`

**Get ESPN WBB Athlete Career Seasons**

## Description

Returns the list of seasons an WBB athlete appeared in. Useful for bounding follow-up calls to per-season endpoints.

## Usage

```
espn_wbb_player_seasons(athlete_id, ...)
```

## Arguments

| | |
|---|---|
| `athlete_id` | ESPN athlete identifier (character or numeric). |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per career season.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| athlete_id | character | ESPN athlete id. |
| season | integer | Season year. |
| ref | character | `$ref` URL to the season detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings()`, `espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`,

espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
# LeBron James (1966): 23 career seasons
espn_wbb_player_seasons(athlete_id = 4433404)
```

---

# espn_wbb_player_stats Get ESPN women's college basketball player stats data

---

## Description

**Get ESPN women's college basketball player stats data**

## Usage

```
espn_wbb_player_stats(athlete_id, year, season_type = "regular", total = FALSE)
```

## Arguments

| | |
|---|---|
| athlete_id | Athlete ID |
| year | Year |
| season_type | (character, default: regular): Season type - regular or postseason |
| total | (boolean, default: FALSE): Totals |

## Value

Returns a tibble with the player stats data

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(),

espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_player_stats(athlete_id = 2984250, year = 2022))
```

---

# espn_wbb_position Get ESPN WBB Position Detail

---

## Description

Returns metadata for a single WBB position. Useful for dereferencing position `$ref` URLs embedded in athlete records, and for navigating parent/leaf relationships in the position taxonomy.

## Usage

```
espn_wbb_position(position_id, ...)
```

## Arguments

| | |
|---|---|
| position_id | ESPN position identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_position(position_id = 1)
```

---

# espn_wbb_positions Get ESPN WBB Positions Index

---

## Description

Returns the WBB position dictionary index. One row per position with its id and the canonical `$ref` URL — pass an id to espn_wbb_position() for full details (display name, abbreviation, leaf flag, parent link).

## Usage

```
espn_wbb_positions(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per position.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_positions()
```

---

# espn_wbb_powerindex Get ESPN WBB Season Power Index (Long Format)

---

## Description

Returns ESPN's Basketball Power Index (BPI) and related per-team metrics for one WBB season, in long format: one row per (team x stat).

## Usage

```
espn_wbb_powerindex(
  season = most_recent_wbb_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WBB season. |
| season_type | Integer (1=preseason, 2=regular (default), 3=postseason). |
| ... | Additional arguments; currently unused. |

## Value

A long tibble with one row per (team x stat).

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_powerindex(season = 2025)
```

---

# espn_wbb_rankings *Get women's college basketball AP and Coaches poll rankings*

---

## Description

Get women's college basketball AP and Coaches poll rankings

## Usage

```
espn_wbb_rankings()
```

## Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
# Get current AP and Coaches Poll rankings

  try(espn_wbb_rankings())
```

---

`espn_wbb_scoreboard` **Get ESPN women's college basketball schedule for a specific year**

---

## Description

**Get ESPN women's college basketball schedule for a specific year**

## Usage

```
espn_wbb_scoreboard(season)
```

## Arguments

| | |
|---|---|
| season | Either numeric or character |

## Value

Returns a tibble

Columns as documented in the shared espn_basketball_scoreboard_schema table.

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
# Get schedule from date 2022-11-15

  try(espn_wbb_scoreboard (season = "20230225"))
```

---

`espn_wbb_season_awards`

# Get ESPN WBB Season Awards Index

---

## Description

Returns the list of award IDs given out in a WBB season from `sports.core.api.espn.com/v2/sports/basketball/leagu` The index only contains IDs and `$ref` URLs — pass an ID to `espn_wbb_award()` for the award name, description, and winners.

## Usage

```
espn_wbb_season_awards(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per award.

| col_name | types | description |
|---|---|---|
| season | integer | Season year. |
| award_id | character | ESPN award identifier. |
| ref | character | Full $ref URL for the award detail. |
| league | character | League slug ("womens-college-basketball"). |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_awards(season = 2024)
```

`espn_wbb_season_group` **Get ESPN WBB Season Group Detail**

## Description

Returns metadata for one group (conference or division) in one (WBB season x season-type), plus `$ref` URLs to its parent group, children groups, member teams, and standings.

Returns the list of child groups (e.g. divisions within a conference) for one (WBB season x season-type x parent-group).

Returns the list of team IDs that belong to one group (conference or division) for one (WBB season x season-type).

## Usage

```
espn_wbb_season_group(
  group_id,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)

espn_wbb_season_group_children(
  group_id,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)

espn_wbb_season_group_teams(
  group_id,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| `group_id` | ESPN group identifier. |
| `season` | Season year. Defaults to most recent WBB season. |
| `season_type` | Season-type id (2 = regular (default)). |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season Group Detail**

**Get ESPN WBB Season Group Children Index**

**Get ESPN WBB Season Group Teams Index**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_group_schema table.

A tibble with one row per child group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| parent_group_id | character | Parent group id (queried). |
| child_group_id | character | Child group id. |
| ref | character | `$ref` to child group detail. |

A tibble with one row per team in the group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| group_id | character | ESPN group id. |
| team_id | character | ESPN team id. |
| ref | character | `$ref` URL to the team-in-season entry. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`, `espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`,

espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_group(group_id = 5, season = 2025)


espn_wbb_season_group_children(group_id = 5, season = 2025)


espn_wbb_season_group_teams(group_id = 5, season = 2025)
```

---

`espn_wbb_season_groups`

# Get ESPN WBB Season Groups Index

---

## Description

Returns the list of group IDs (conferences / divisions) for one (WBB season x season-type) via core-v2 `/seasons/{season}/types/{season_type}/groups`.

## Usage

```
espn_wbb_season_groups(
  season = most_recent_wbb_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| `season` | Season year. Defaults to most recent WBB season. |
| `season_type` | Season-type id (2 = regular (default)). |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| group_id | character | ESPN group id. |
| ref | character | `$ref` URL for the group detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_groups(season = 2025)
```

---

`espn_wbb_season_info` **Get ESPN Women's College Basketball Season Info**

---

## Description

**Get ESPN Women's College Basketball Season Info**

**Get ESPN Women's College Basketball Season Info**

## Usage

```
espn_wbb_season_info(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric, e.g. 2025). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use options(wehoop.proxy = ...) – see ?wehoop for details. |

## Value

A named list of wehoop_data tibbles: Info, Types, Athletes, Coaches, Teams, Awards. $ref URL components are returned as character columns and are NOT auto-resolved – use targeted endpoint functions for details.

### Info

| col_name | types | description |
|---|---|---|
| year | integer | 4-digit year. |
| start_date | character | Start date (YYYY-MM-DD). |
| end_date | character | End date (YYYY-MM-DD). |
| display_name | character | Display name. |
| type_id | character | Type identifier (numeric). |
| type_name | character | Type name. |

### Types / Athletes / Coaches / Teams / Awards

| col_name | types | description |
|---|---|---|
| count | integer | Count of count. |
| ref | character | Ref. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(),

espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_info(season = 2025)
```

---

`espn_wbb_season_leaders`

# Get ESPN WBB Season Leaders (Long Format)

---

## Description

Returns the per-category leaderboard for one (WBB season x season-type), in long format. Each row is one (category x rank) entry, e.g. "Points Per Game x rank 1 x LeBron James".

## Usage

```
espn_wbb_season_leaders(
  season = most_recent_wbb_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year. Defaults to most recent WBB season. |
| season_type | Season-type id (2 = regular (default), 3 = postseason). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season Leaders (Long Format)**

## Value

A long tibble with one row per (category x leader).

Columns as documented in the shared espn_basketball_season_leaders_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_leaders(season = 2025)
```

---

`espn_wbb_season_ranking`

# Get ESPN WBB Season Ranking Detail

---

## Description

Returns the per-week snapshot index for one ranking source (e.g. AP Top 25). Each row is one weekly snapshot; the `ref` URL resolves to the actual ranked teams for that (season-type x week) and will be wrapped by a forthcoming `espn_LOWERWBB_week_ranking()`.

## Usage

```
espn_wbb_season_ranking(ranking_id, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| `ranking_id` | ESPN ranking identifier (character or numeric). |
| `season` | Season year. Defaults to most recent WBB season. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season Ranking Detail**

## Value

A tibble with one row per weekly snapshot.

Columns as documented in the shared espn_basketball_season_ranking_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_ranking(ranking_id = 1, season = 2025)
```

---

`espn_wbb_season_rankings`

**Get ESPN WBB Season Rankings Index**

---

## Description

Returns the index of season-level rankings recorded for one WBB season. Typical sources: AP Top 25 (id 1), Coaches Poll (id 2). Pass an `ranking_id` to `espn_wbb_season_ranking()` for the ranked teams.

## Usage

```
espn_wbb_season_rankings(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year. Defaults to most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season Rankings Index**

## Value

A tibble with one row per ranking source.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| ranking_id | character | ESPN ranking id. |
| ref | character | $ref URL for the ranking detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_rankings(season = 2025)
```

---

`espn_wbb_season_type` **Get ESPN WBB Season-Type Detail**

---

## Description

Returns metadata for one season-type within an WBB season: name, abbreviation, start / end dates, and whether it carries groups, standings, or playoff legs.

## Usage

```
espn_wbb_season_type(season_type = 2L, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| `season_type` | Season-type id (1 = preseason, 2 = regular (default), 3 = postseason, 4 = off-season). |
| `season` | Season year. Defaults to most recent WBB season. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season-Type Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_type_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`,

espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_type(season_type = 2, season = 2025)
```

---

# espn_wbb_season_types Get ESPN WBB Season Types Index

---

## Description

Returns the index of season-type IDs that exist for one WBB season (typically 1 = preseason, 2 = regular, 3 = postseason, 4 = off-season). Pass an ID to espn_wbb_season_type() for the start/end dates and whether that type carries groups, standings, or legs.

## Usage

```
espn_wbb_season_types(season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WBB season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per season type.

| col_name | types | description |
|---|---|---|
| league | character | League slug ("womens-college-basketball"). |
| season | integer | Season year. |
| season_type | integer | Season-type id (1/2/3/4). |
| ref | character | $ref URL for the type detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_types(season = 2025)
```

---

# espn_wbb_season_week Get ESPN WBB Season-Week Detail

---

## Description

Returns metadata for one week (number, start / end dates, text label, and `$ref` to the per-week rankings endpoint).

## Usage

```
espn_wbb_season_week(
  week,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| week | Week number. |
| season | Season year. Defaults to most recent WBB season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Season-Week Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_week_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_season_week(week = 5, season = 2025)
```

---

# espn_wbb_season_weeks **Get ESPN WBB Season Weeks Index**

---

## Description

Returns the list of week IDs for one (WBB season x season-type). WBB uses a week structure inherited from ESPN's schema, but week-level rankings are populated only for college (MBB / WBB).

## Usage

```
espn_wbb_season_weeks(
  season = most_recent_wbb_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year. Defaults to most recent WBB season. |
| season_type | Season-type id (2 = regular (default), 3 = postseason). |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per week.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| week | integer | Week number (1-based). |
| ref | character | $ref URL for the week detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule() espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

### Examples

```
espn_wbb_season_weeks(season = 2025)
```

---

`espn_wbb_seasons` **Get ESPN Women's College Basketball Seasons**

---

### Description

**Get ESPN Women's College Basketball Seasons**

**Get ESPN Women's College Basketball Seasons**

### Usage

```
espn_wbb_seasons(...)
```

### Arguments

| | |
|---|---|
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

### Value

A single `wehoop_data` tibble with one row per season.

| col_name | types | description |
|---|---|---|
| season | integer | Season identifier (4-digit year or 'YYYY-YY' string). |
| start_date | character | Start date (YYYY-MM-DD). |
| end_date | character | End date (YYYY-MM-DD). |
| display_name | character | Display name. |
| season_type_count | integer | Count of season type. |

### Author(s)

Saiem Gilani

### See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`,

espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_seasons()
```

---

# espn_wbb_standings *Get ESPN women's college basketball standings*

---

## Description

Get ESPN women's college basketball standings

## Usage

```
espn_wbb_standings(year)
```

## Arguments

| | |
|---|---|
| year | Either numeric or character (YYYY) |

## Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(),

espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_standings(2021))
```

---

# espn_wbb_team — Get ESPN Women's College Basketball Team Detail

---

## Description

Era-correct team identity for a women's college basketball program in a specific season, plus the available `$ref` URLs for deeper resources (record, statistics, leaders, athletes/roster, coaches, college, etc.). Backed by the core-v2 endpoint `sports.core.api.espn.com/v2/sports/basketball/leagues/womens-coll`

Historical depth goes back to **1982** at ESPN (NCAA-tournament-era forward). Older seasons return fewer `$ref` keys; missing refs become `NA`.

## Usage

```
espn_wbb_team(team_id, season = most_recent_wbb_season(), ...)

espn_wbb_team_roster(team_id, season = most_recent_wbb_season(), ...)

espn_wbb_team_leaders(team_id, season = most_recent_wbb_season(), ...)

espn_wbb_team_season_profile(team_id, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier (character or numeric). |
| `season` | Season year (numeric). Defaults to the most recent WBB season. |
| `...` | Additional arguments; currently unused. |

## Value

A named list of data frames: `Info`, `Record`, `NextEvent`, `StandingSummary`, `Coaches`.

**Info**

Columns as documented in the shared espn_basketball_team_info_schema table.

**Record**

| col_name | types | description |
|---|---|---|
| type | character | Record type / category. |
| summary | character | Summary. |
| stats | list | Stats. |

**NextEvent**

Columns as documented in the shared espn_basketball_player_overview_next_game_schema table.

**StandingSummary**

| col_name | types | description |
|---|---|---|
| standing_summary | character | Standing summary. |

**Coaches**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| first_name | character | Player's first name. |
| last_name | character | Player's last name. |
| experience | integer | Years of professional experience. |

A single tibble with one row per athlete.

Columns as documented in the shared espn_basketball_team_roster_schema table. ESPN's roster endpoint ignores the season, so a season before the current one is built from that season's published player box scores (name, jersey, position and headshot from the athlete's last game; bio columns NA), and errors when none are published.

A single long-format tibble (one row per category-rank-athlete).

Columns as documented in the shared espn_basketball_team_leaders_schema table.

A single-row tibble with team identity scalars and _ref URL columns. Selected columns:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(),

espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team(team_id = "2509", season = 2025)

espn_wbb_team_roster(team_id = "2509", season = 2025)

espn_wbb_team_leaders(team_id = "2509", season = 2025)

espn_wbb_team_season_profile(team_id = "2509", season = 2025)
```

---

`espn_wbb_team_injuries`

# Get ESPN Women's College Basketball Team Injuries

---

## Description

**Get ESPN Women's College Basketball Team Injuries**

**Get ESPN Women's College Basketball Team Injuries**

## Usage

```
espn_wbb_team_injuries(team_id, ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier (character or numeric; passed as-is). Use `espn_wbb_teams()` to look up team IDs. Example: `"2509"` (UConn). |
| `...` | Currently unused; reserved for future argument threading. |

## Value

Returns a tibble of injury records for the specified WBB team. Returns an empty tibble (zero rows) when the team has no reported injuries.

### Injuries

Columns as documented in the shared espn_basketball_team_injuries_injuries_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_injuries(team_id = "2509")
```

---

# espn_wbb_team_news Get ESPN WBB Team News

---

## Description

**Get ESPN WBB Team News**

**Get ESPN Women's College Basketball Team News**

## Usage

```
espn_wbb_team_news(team_id, limit = 25)
```

## Arguments

| | |
|---|---|
| team_id | character or integer. ESPN team ID (e.g. 2509 for UConn). |
| limit | integer. Maximum number of articles to return. Default 25. |

## Details

Retrieve ESPN women's college basketball news for a specific team. Uses getOption("wehoop.proxy") or http_proxy/https_proxy environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of team news articles.

Columns as documented in the shared espn_basketball_news_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_news(team_id = "2509", limit = 5)
```

espn_wbb_team_record_detail

**Get ESPN WBB Team Record Detail (Long Format)**

## Description

Returns one team's record detail in long format: one row per stat in the record's `stats[]` array. Use the record-index endpoint to enumerate available `record_id` values per team-season (overall / home / away / conference + per-opponent breakdowns).

## Usage

```
espn_wbb_team_record_detail(team_id, season, record_id, season_type = 2L, ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year (numeric). |
| `record_id` | Record identifier (from the record-index endpoint index). |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_schedule(),

espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_record_detail(team_id = 2509, season = 2024, record_id = 0)
```

---

espn_wbb_team_schedule

# Get ESPN Women's College Basketball Team Schedule

---

## Description

**Get ESPN Women's College Basketball Team Schedule**

**Get ESPN Women's College Basketball Team Schedule**

## Usage

```
espn_wbb_team_schedule(
  team_id,
  season = most_recent_wbb_season(),
  season_type = 2,
  ...
)
```

## Arguments

| | |
|---|---|
| team_id | ESPN team identifier (character or numeric). |
| season | Season year (numeric). Defaults to the most recent WBB season. |
| season_type | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| ... | Additional arguments; currently unused. |

## Value

A single tibble with one row per event.

Columns as documented in the shared espn_basketball_team_schedule_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_schedule(team_id = "2509", season = 2025)
```

---

`espn_wbb_team_season_roster`

# Get ESPN WBB Team Roster (Per-Season, core-v2)

---

## Description

Returns the per-season team roster as a tibble of athlete IDs from `seasons/{y}/teams/{id}/athletes`. Distinct from `espn_wbb_team_roster()` which targets a site-v2 endpoint optimized for the current season; this core-v2 variant is era-correct and available back to ESPN's earliest season for each league.

## Usage

```
espn_wbb_team_season_roster(team_id, season = most_recent_wbb_season(), ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year. Defaults to most recent WBB season. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Team Roster (Per-Season, core-v2)**

## Value

A tibble with one row per athlete on the season roster.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| team_id | character | ESPN team id. |
| season | integer | Season year. |
| athlete_id | character | ESPN athlete id. |
| ref | character | `$ref` URL to athlete-in-season detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_season_roster(team_id = 2509, season = 2025)
```

`espn_wbb_team_season_statistics` **Get ESPN WBB Team Season Statistics (Long Format with Rank)**

## Description

Returns the full team-season-type statistics sheet for one WBB team in long format: one row per (category x stat). Each row carries the team's league rank for that stat where ESPN provides it (`rank` + `rank_display_value`). Complements W-L records (not directly wrapped for WBB) with the full stat package.

## Usage

```
espn_wbb_team_season_statistics(
  team_id,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year (numeric). Defaults to the most recent WBB season. |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble with one row per (category x stat).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`,

espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_team_season_statistics(team_id = 2509, season = 2024)
```

---

# espn_wbb_team_stats Get ESPN women's college basketball team stats data

---

## Description

**Get ESPN women's college basketball team stats data**

## Usage

```
espn_wbb_team_stats(team_id, year, season_type = "regular", total = FALSE)
```

## Arguments

| | |
|---|---|
| team_id | Team ID |
| year | Year |
| season_type | (character, default: regular): Season type - regular or postseason |
| total | (boolean, default: FALSE): Totals |

## Value

Returns a tibble with the team stats data

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_team_stats(team_id = 2509, year = 2020))
```

---

# espn_wbb_teams *Get ESPN women's college basketball team names and ids*

---

## Description

Get ESPN women's college basketball team names and ids

## Usage

```
espn_wbb_teams(year = most_recent_wbb_season())
```

## Arguments

| | |
|---|---|
| year | Either numeric or character (YYYY) |

## Value

Returns a teams infomation data frame

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
try(espn_wbb_teams())
```

---

# espn_wbb_tournament Get ESPN WBB Tournament Detail

---

## Description

Returns metadata for a single WBB tournament plus the `$ref` URL for the tournament's seasons list. Tournament IDs of interest include the NCAA Tournament and WNIT (resolve via espn_wbb_tournaments()).

## Usage

```
espn_wbb_tournament(tournament_id, ...)
```

## Arguments

| | |
|---|---|
| tournament_id | ESPN tournament identifier. |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

| col_name | types | description |
|---|---|---|
| tournament_id | character | ESPN tournament identifier. |
| display_name | character | Human-readable tournament name. |
| seasons_ref | character | $ref to the seasons-list endpoint. |
| league | character | League slug. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_tournament(tournament_id = 3)
```

---

`espn_wbb_tournament_season`

# Get ESPN WBB Tournament Season Detail

---

## Description

Returns single-row detail for one (tournament, season) pair: id, display name, number of rounds, and $refs to the season + bracketology resources. Use espn_wbb_tournament_seasons() to enumerate valid (tournament_id, season) pairs.

## Usage

```
espn_wbb_tournament_season(tournament_id, season, ...)
```

### Arguments

| | |
|---|---|
| tournament_id | ESPN tournament identifier. |
| season | Season year (numeric). |
| ... | Additional arguments; currently unused. |

### Value

A single-row tibble.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

### Examples

```
espn_wbb_tournament_season(tournament_id = 3, season = 2024)
```

---

## espn_wbb_tournament_seasons

**Get ESPN WBB Tournament Seasons List**

---

### Description

Returns the seasons in which a given WBB tournament was held.

## Usage

```
espn_wbb_tournament_seasons(tournament_id, ...)
```

## Arguments

| | |
|---|---|
| tournament_id | ESPN tournament identifier. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per season.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| tournament_id | character | ESPN tournament identifier. |
| season | integer | Season year. |
| ref | character | Full $ref URL for that season. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_tournament_seasons(tournament_id = 3)
```

`espn_wbb_tournaments` **Get ESPN WBB Tournaments Index**

## Description

Returns the index of women's college basketball tournaments tracked by ESPN — including the NCAA Tournament, WNIT, conference tournaments, and other in-season events. Pass an ID to `espn_wbb_tournament()` for the human-readable name and seasons-list ref.

Note: ESPN does not expose a tournaments endpoint for the WNBA.

## Usage

```
espn_wbb_tournaments(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per tournament.

| col_name | types | description |
|---|---|---|
| tournament_id | character | ESPN tournament identifier. |
| ref | character | Full `$ref` URL for the detail. |
| league | character | League slug (`"womens-college-basketball"`). |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: `espn_wbb_athletes_index()`, `espn_wbb_award()`, `espn_wbb_calendar()`, `espn_wbb_coach()`, `espn_wbb_coach_record()`, `espn_wbb_coach_season()`, `espn_wbb_coaches()`, `espn_wbb_conferences()`, `espn_wbb_franchise()`, `espn_wbb_franchises()`, `espn_wbb_game_all()`, `espn_wbb_game_odds()`, `espn_wbb_game_official_detail()`, `espn_wbb_game_play()`, `espn_wbb_game_play_personn`
`espn_wbb_game_player_box()`, `espn_wbb_game_probabilities()`, `espn_wbb_game_propbets()`, `espn_wbb_game_team_linescores()`, `espn_wbb_game_team_roster_entry()`, `espn_wbb_injuries()`, `espn_wbb_leaders()`, `espn_wbb_news()`, `espn_wbb_player_awards()`, `espn_wbb_player_career_stats()`, `espn_wbb_player_eventlog_v2()`, `espn_wbb_player_info()`, `espn_wbb_player_overview()`, `espn_wbb_player_seasons()`, `espn_wbb_player_stats()`, `espn_wbb_position()`, `espn_wbb_positions()`, `espn_wbb_powerindex()`, `espn_wbb_rankings()`, `espn_wbb_scoreboard()`, `espn_wbb_season_awards()`, `espn_wbb_season_group()`, `espn_wbb_season_groups()`, `espn_wbb_season_info()`, `espn_wbb_season_leaders()`, `espn_wbb_season_ranking()`, `espn_wbb_season_rankings()`, `espn_wbb_season_type()`, `espn_wbb_season_types()`, `espn_wbb_season_week()`, `espn_wbb_season_weeks()`, `espn_wbb_seasons()`, `espn_wbb_standings()`,

espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_venues(), espn_wbb_week_ranking(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_tournaments()
```

---

espn_wbb_venues **Get ESPN Women's College Basketball Venues**

---

## Description

**Get ESPN Women's College Basketball Venues**

**Get ESPN Women's College Basketball Venues**

## Usage

```
espn_wbb_venues(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see ?wehoop for details. |

## Value

A single `wehoop_data` tibble with one row per venue.

Columns as documented in the shared espn_basketball_venues_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(),

espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_week_ranking(), espn_wbb_week_rankings(

## Examples

```
espn_wbb_venues()
```

---

# espn_wbb_week_ranking Get ESPN WBB Per-Week Ranking Detail

---

## Description

Returns the long-format ranked teams for one (season x season-type x week x ranking-source). Typically 25 rows for WBB (AP Top 25).

## Usage

```
espn_wbb_week_ranking(
  ranking_id,
  week,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| ranking_id | Ranking source id (1 = AP, 2 = Coaches, etc.). |
| week | Week number. |
| season | Season year. Defaults to most recent WBB season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Per-Week Ranking Detail**

## Value

A tibble with one row per ranked team (typically 25).

Columns as documented in the shared espn_basketball_week_ranking_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_rankings()

## Examples

```
espn_wbb_week_ranking(ranking_id = 1, week = 5, season = 2025)
```

---

espn_wbb_week_rankings

# Get ESPN WBB Per-Week Rankings Index

---

## Description

Returns the index of ranking sources available for one (WBB season x season-type x week). Typical sources: AP Top 25 (id 1), Coaches Poll (id 2). Pass an id to espn_wbb_week_ranking() for the ranked teams.

## Usage

```
espn_wbb_week_rankings(
  week,
  season = most_recent_wbb_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| week | Week number. |
| season | Season year. Defaults to most recent WBB season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WBB Per-Week Rankings Index**

## Value

A tibble with one row per ranking source.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| week | integer | Week number. |
| ranking_id | character | ESPN ranking id. |
| ref | character | $ref URL for the ranked-teams detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WBB Functions: espn_wbb_athletes_index(), espn_wbb_award(), espn_wbb_calendar(), espn_wbb_coach(), espn_wbb_coach_record(), espn_wbb_coach_season(), espn_wbb_coaches(), espn_wbb_conferences(), espn_wbb_franchise(), espn_wbb_franchises(), espn_wbb_game_all(), espn_wbb_game_odds(), espn_wbb_game_official_detail(), espn_wbb_game_play(), espn_wbb_game_play_personn espn_wbb_game_player_box(), espn_wbb_game_probabilities(), espn_wbb_game_propbets(), espn_wbb_game_team_linescores(), espn_wbb_game_team_roster_entry(), espn_wbb_injuries(), espn_wbb_leaders(), espn_wbb_news(), espn_wbb_player_awards(), espn_wbb_player_career_stats(), espn_wbb_player_eventlog_v2(), espn_wbb_player_info(), espn_wbb_player_overview(), espn_wbb_player_seasons(), espn_wbb_player_stats(), espn_wbb_position(), espn_wbb_positions(), espn_wbb_powerindex(), espn_wbb_rankings(), espn_wbb_scoreboard(), espn_wbb_season_awards(), espn_wbb_season_group(), espn_wbb_season_groups(), espn_wbb_season_info(), espn_wbb_season_leaders(), espn_wbb_season_ranking(), espn_wbb_season_rankings(), espn_wbb_season_type(), espn_wbb_season_types(), espn_wbb_season_week(), espn_wbb_season_weeks(), espn_wbb_seasons(), espn_wbb_standings(), espn_wbb_team(), espn_wbb_team_injuries(), espn_wbb_team_news(), espn_wbb_team_record_detail(), espn_wbb_team_schedule(), espn_wbb_team_season_roster(), espn_wbb_team_season_statistics(), espn_wbb_team_stats(), espn_wbb_teams(), espn_wbb_tournament(), espn_wbb_tournament_season(), espn_wbb_tournament_seasons(), espn_wbb_tournaments(), espn_wbb_venues(), espn_wbb_week_ranking()

## Examples

```
espn_wbb_week_rankings(week = 5, season = 2025)
```

---

espn_wnba_athletes_index

# Get ESPN WNBA Athletes Index

---

## Description

**Get ESPN WNBA Athletes Index**

**Get ESPN WNBA Athletes Index**

## Usage

```
espn_wnba_athletes_index(
  season = most_recent_wnba_season(),
  active = TRUE,
  limit = 5000L,
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| active | logical. When TRUE (default) only active athletes are returned. Set to FALSE for the full historical roster. |
| limit | integer. Maximum number of rows to return. Default 5000. Pass a small value (e.g. limit = 50) in tests to keep execution fast. |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use options(wehoop.proxy = ...) – see ?wehoop for details. |

## Value

A single wehoop_data tibble with one row per athlete.

Columns as documented in the shared espn_basketball_athletes_index_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_athletes_index(season = 2025, limit = 50)
```

---

# espn_wnba_award Get ESPN WNBA Season Award Detail

---

## Description

Returns the name, description, and winners of one WNBA season award.

## Usage

```
espn_wnba_award(award_id, season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| award_id | ESPN award identifier (character or numeric). |
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per winner.

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

### Examples

```
espn_wnba_award(award_id = 247, season = 2024)
```

---

## espn_wnba_calendar  Get ESPN WNBA Calendar

---

### Description

**Get ESPN WNBA Calendar**

**Get ESPN WNBA Calendar**

### Usage

```
espn_wnba_calendar(season = most_recent_wnba_season())
```

### Arguments

| | |
|---|---|
| season | integer or character. Four-digit season year (e.g. 2025). Defaults to most_recent_wnba_season(). |

## Details

Retrieve the ESPN WNBA schedule calendar for a given season. The underlying scoreboard response contains a `leagues[[1]]$calendar` block with season-type entries (pre-season, regular, post). Uses `getOption("wehoop.proxy")` or `http_proxy`/`https_proxy` environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of calendar entries.

Columns as documented in the shared espn_basketball_calendar_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_calendar(season = 2025)
```

---

`espn_wnba_coach_season`

# Get ESPN WNBA Coach-in-Season Detail

---

## Description

Per-season coach detail (name, birth info, $refs to team/college/ person). ESPN's coverage of this endpoint is sparse; many (coach × season) combinations return 404.

## Usage

```
espn_wnba_coach_season(coach_id, season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| coach_id | ESPN coach identifier. |
| season | Season year. Defaults to most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Coach-in-Season Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_coach_season_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankin espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_coach_season(coach_id = 52063, season = 2025)
```

---

`espn_wnba_coaches` **Get ESPN WNBA Coaches**

---

## Description

**Get ESPN WNBA Coaches**

**Get ESPN WNBA Coaches**

## Usage

```
espn_wnba_coaches(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single `wehoop_data` tibble with one row per coach.

| col_name | types | description |
|---|---|---|
| coach_id | character | Unique identifier for coach. |
| first_name | character | Player's first name. |
| last_name | character | Player's last name. |
| full_name | character | Player's full name. |
| experience | integer | Years of professional experience. |
| team_id | character | Unique team identifier. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_coaches(season = 2025)
```

---

# espn_wnba_conferences Get ESPN WNBA Conferences

---

## Description

**Get ESPN WNBA Conferences**

**Get ESPN WNBA Conferences**

## Usage

```
espn_wnba_conferences()
```

## Value

A wehoop_data tibble with one row per conference:

Columns as documented in the shared espn_basketball_conferences_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_conferences())
```

---

`espn_wnba_draft` **Get ESPN WNBA Draft Picks**

---

## Description

**Get ESPN WNBA Draft Picks**

**Get ESPN WNBA Draft Picks**

## Usage

```
espn_wnba_draft(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)`. |

## Details

Calls the ESPN core-v2 endpoint `https://sports.core.api.espn.com/v2/sports/basketball/leagues/wnba/seasor` which returns each round of the draft with its picks inlined as `picks: [...]`. For historical seasons with no ESPN draft data the function returns an empty tibble rather than erroring.

## Value

A wehoop_data tibble with one row per draft pick:

See the HTML help or pkgdown reference for the column table.

Athlete and team details (name, position, college, abbreviation) are not inlined in the draft response; resolve them via espn_wnba_player_info() or espn_wnba_team() using the returned IDs.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft_athlete_detail( espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankin espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_draft(season = 2024)
```

---

# espn_wnba_draft_athlete_detail

**Get ESPN WNBA Draft Athlete Detail (Single Drafted Player)**

---

## Description

Returns rich single-row detail for one drafted athlete in one NBA draft year: name, height, weight, position, pick (overall/round/team), and a $ref to the athlete's core-v2 profile. Use espn_wnba_draft_athletes() to enumerate draftees for a year.

## Usage

```
espn_wnba_draft_athlete_detail(
  season = most_recent_wnba_season(),
  athlete_id,
  ...
)
```

## Arguments

| | |
|---|---|
| season | Draft year (numeric). Defaults to the most recent NBA season. |
| athlete_id | ESPN draftee identifier. |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_draft_athlete_detail(season = 2024, athlete_id = 3149391)
```

`espn_wnba_draft_pick` **Get ESPN WNBA Draft Pick Detail**

## Description

Returns a single WNBA draft pick. Defaults to the most recent WNBA season's #1 overall pick. For a full draft, use `espn_wnba_draft()`.

## Usage

```
espn_wnba_draft_pick(
  season = most_recent_wnba_season(),
  round = 1L,
  pick = 1L,
  ...
)
```

## Arguments

| | |
|---|---|
| `season` | Season year. Defaults to most recent WNBA season. |
| `round` | Draft round (default 1). |
| `pick` | Pick number within the round (default 1). |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Draft Pick Detail**

## Value

A single-row tibble.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_per`
`espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`,

espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_draft_pick(season = 2024, round = 1, pick = 1)
```

---

espn_wnba_draft_rounds

# Get ESPN WNBA Draft Rounds Summary

---

## Description

Returns one row per round of the WNBA draft (typically 2 rounds: 1st with 30 picks, 2nd with ~28 picks).

Returns one row per drafted athlete in a given WNBA draft year.

Returns the current status of one WNBA draft year (round, state, description). Live during the draft; static afterward.

Returns a single-row tibble with top-level draft-year metadata: year, number of rounds, display name, plus `$ref`s for the deeper sub-resources (status, athletes, rounds) already wrapped by espn_wnba_draft_status(), espn_wnba_draft_athletes(), and espn_wnba_draft_rounds().

## Usage

```
espn_wnba_draft_rounds(season = most_recent_wnba_season(), ...)

espn_wnba_draft_athletes(season = most_recent_wnba_season(), ...)

espn_wnba_draft_status(season = most_recent_wnba_season(), ...)

espn_wnba_season_draft(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Draft Rounds Summary**

**Get ESPN WNBA Draft Athletes Index**

**Get ESPN WNBA Draft Status**

## Value

A tibble with one row per round.

A tibble of athlete IDs + `$ref` URLs.

A single-row tibble.

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankin espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_draft_rounds(season = 2024)
```

```
espn_wnba_draft_athletes(season = 2024)
```

```
espn_wnba_draft_status(season = 2024)
```

```
espn_wnba_season_draft(season = 2024)
```

---

`espn_wnba_franchise` **Get ESPN WNBA Franchise Detail**

---

## Description

Returns franchise-level metadata for a WNBA franchise. Franchise IDs are stable across relocations and rebrands.

## Usage

```
espn_wnba_franchise(franchise_id, ...)
```

## Arguments

| | |
|---|---|
| `franchise_id` | ESPN franchise identifier (character or numeric). |
| `...` | Additional arguments; currently unused. |

## Value

A single-row tibble.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_per espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_franchise(franchise_id = 17)
```

---

`espn_wnba_franchises` **Get ESPN WNBA Franchises Index**

---

## Description

Returns the full WNBA franchises index from `sports.core.api.espn.com/v2/sports/basketball/leagues/wnba/fran`
Each row is one franchise with its ID and the canonical `$ref` URL — pass an ID to `espn_wnba_franchise()`
for full franchise detail.

## Usage

```
espn_wnba_franchises(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per franchise.

| col_name | types | description |
|---|---|---|
| franchise_id | character | ESPN franchise identifier. |
| ref | character | Full `$ref` URL for franchise detail. |
| league | character | League slug ("wnba"). |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_per` `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`, `espn_wnba_position()`, `espn_wnba_positions()`,

espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_franchises()
```

---

`espn_wnba_freeagents` **Get ESPN WNBA Free Agents**

---

## Description

**Get ESPN WNBA Free Agents**

**Get ESPN WNBA Free Agents**

## Usage

```
espn_wnba_freeagents(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)`. |

## Details

Calls the ESPN core-v2 endpoint `https://sports.core.api.espn.com/v2/sports/basketball/leagues/wnba/seasor` As of 2026-05, this endpoint returns HTTP 500 for every season tested and there is no documented replacement; the function consequently returns an empty tibble and emits a CLI error message. It is retained so that downstream code does not break if ESPN restores the endpoint, but should not be relied on in the meantime.

## Value

A `wehoop_data` tibble with one row per free agent:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_per espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_freeagents(season = 2025)
```

---

`espn_wnba_futures` **Get ESPN WNBA Season Futures (Long Format)**

---

## Description

Returns the full futures-betting board for a WNBA season, in long format: one row per (market x team). Markets include championship winner and conference winners. Backed by `sports.core.api.espn.com/v2/sports`.

Note: ESPN does not currently expose a futures endpoint for women's college basketball.

## Usage

```
espn_wnba_futures(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric). Defaults to the most recent WNBA season. |
| `...` | Additional arguments; currently unused. |

## Value

A long-format tibble.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_per espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_futures(season = 2025)
```

---

`espn_wnba_game_all` *Get ESPN's WNBA game data (play-by-play, team and player box)*

---

## Description

Get ESPN's WNBA game data (play-by-play, team and player box)

Get ESPN's WNBA play by play data

Get ESPN's WNBA team box data

Get ESPN's WNBA player box data

**Get ESPN WNBA game rosters**

## Usage

```
espn_wnba_game_all(game_id)

espn_wnba_pbp(game_id)
```

```
espn_wnba_team_box(game_id)

espn_wnba_player_box(game_id)

espn_wnba_game_rosters(game_id)
```

## Arguments

| | |
|---|---|
| game_id | Game ID |

## Value

A named list of dataframes: Plays, Team, Player

**Plays**

Columns as documented in the shared espn_basketball_game_all_plays_schema table.

**Team**

Columns as documented in the shared espn_basketball_team_box_schema table.

**Player**

Columns as documented in the shared espn_basketball_game_all_player_schema table.

Returns a play-by-play data frame

**Plays**

Columns as documented in the shared espn_basketball_game_all_plays_schema table.

Returns a team boxscore data frame

**Team**

Columns as documented in the shared espn_basketball_team_box_schema table.

Returns a player boxscore data frame

**Player**

Columns as documented in the shared espn_basketball_game_all_player_schema table.

A game rosters data frame

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_per espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(),

espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_game_all(game_id = 401244185))


 try(espn_wnba_pbp(game_id = 401455681))


 try(espn_wnba_team_box(game_id = 401244185))

try(espn_wnba_player_box(game_id = 401244185))


try(espn_wnba_game_rosters(game_id = 401244185))
```

---

espn_wnba_game_odds **Get ESPN WNBA Event Odds**

---

## Description

Returns the live game situation for one WNBA event: timeouts remaining, team fouls, fouls to give, bonus state, and a `$ref` to the last play. During a live game this reflects current state; after the game ends the values are frozen.

Returns pre-game predictor statistics for one WNBA event in long format: one row per (team × statistic). Typical stats include matchup quality, predicted score, win probability, and team strength metrics. Returns empty for events without predictor data (often the case for already-played games).

Returns the per-team power-index `$ref` URLs for one WNBA event. Coverage is sparse — many events return zero items.

## Usage

```
espn_wnba_game_odds(event_id, ...)

espn_wnba_game_officials(event_id, ...)

espn_wnba_game_broadcasts(event_id, ...)

espn_wnba_game_situation(event_id, ...)

espn_wnba_game_predictor(event_id, ...)

espn_wnba_game_powerindex(event_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Event Predictor (Pre-game)**

**Get ESPN WNBA Event Power Index Index**

## Value

A tibble with one row per odds provider.

Columns as documented in the shared espn_basketball_game_odds_schema table.

A tibble with one row per official assigned to the game.

Columns as documented in the shared espn_basketball_game_officials_schema table.

A tibble with one row per broadcast outlet for the game.

Columns as documented in the shared espn_basketball_game_broadcasts_schema table.

A single-row tibble with timeouts + fouls for both teams.

A long tibble with rows for both home and away teams.

A tibble with one row per team-game power-index entry.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_pers

espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_odds(event_id = "401736171")


espn_wnba_game_officials(event_id = "401736171")


espn_wnba_game_broadcasts(event_id = "401736171")


espn_wnba_game_situation(event_id = 401736171)


espn_wnba_game_predictor(event_id = 401736171)


espn_wnba_game_powerindex(event_id = 401736171)
```

---

`espn_wnba_game_official_detail`

# Get ESPN WNBA Event Official Detail (Single Official)

---

## Description

Returns a single-row tibble for one referee assigned to one WNBA event, with their name, position (Referee / Crew Chief / Umpire), and crew order. Pair with espn_wnba_game_officials() to enumerate the crew.

## Usage

```
espn_wnba_game_official_detail(event_id, order, ...)
```

### Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| order | Crew order index (1 = first official). Pair with the order column from event_officials(). |
| ... | Additional arguments; currently unused. |

### Value

A single-row tibble.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

### Examples

```
espn_wnba_game_official_detail(event_id = 401736171, order = 1)
```

---

## espn_wnba_game_play  Get ESPN WNBA Event Play Detail (Single Play)

---

### Description

Returns the rich detail block for a single WNBA play: sequence, period, clock, text, scoring/shooting flags, current home/away score, team $ref, and shot coordinates if applicable. Complements the bulk espn_wnba_pbp() output by exposing the canonical core-v2 play record.

## Usage

```
espn_wnba_game_play(event_id, play_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| play_id | ESPN play identifier (visible in pbp() output as play_id or extractable from plays[].$ref). |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play_pers espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_play(event_id = 401736171, play_id = 4017361714)
```

`espn_wnba_game_play_personnel` **Get ESPN WNBA Event Play Personnel (On-Court Lineup at Play)**

## Description

Returns the players on court at a specific WNBA play in long format (one row per athlete entry across both competitors). Foundation for lineup analysis. ESPN coverage is sparse — many plays return zero rows; the wrapper returns a typed empty tibble in that case.

## Usage

```
espn_wnba_game_play_personnel(event_id, play_id, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `play_id` | ESPN play identifier. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble with one row per on-court athlete.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`, `espn_wnba_position()`, `espn_wnba_positions()`, `espn_wnba_powerindex()`, `espn_wnba_scoreboard()`, `espn_wnba_season_awards()`, `espn_wnba_season_group()`, `espn_wnba_season_groups()`, `espn_wnba_season_info()`, `espn_wnba_season_leaders()`, `espn_wnba_season_rankin`
`espn_wnba_season_rankings()`, `espn_wnba_season_type()`, `espn_wnba_season_types()`, `espn_wnba_season_week()`
`espn_wnba_season_weeks()`, `espn_wnba_seasons()`, `espn_wnba_standings()`, `espn_wnba_team()`, `espn_wnba_team_injuries()`, `espn_wnba_team_news()`, `espn_wnba_team_record()`, `espn_wnba_team_record_detail`
`espn_wnba_team_schedule()`, `espn_wnba_team_season_roster()`, `espn_wnba_team_season_statistics()`,

espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_play_personnel(event_id = 401736171, play_id = 4017361714)
```

---

`espn_wnba_game_player_box`

# Get ESPN WNBA Event Player Box Score (Long Format)

---

## Description

Returns the long-format per-game box score for a single athlete in one WNBA event. One row per (category x stat). Same shape as `espn_wnba_game_team_statistics()` but scoped to a single athlete-in-event instead of the full team. `stat_type` defaults to 0 (regular-season aggregate as ESPN tags it for finished events).

## Usage

```
espn_wnba_game_player_box(event_id, team_id, athlete_id, stat_type = 0L, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `team_id` | ESPN team identifier (the competitor the athlete played for). |
| `athlete_id` | ESPN athlete identifier. |
| `stat_type` | Integer stat-type segment. Defaults to 0 (the only type commonly populated for finished events). |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble with one row per (category x stat).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_player_box(event_id = 401736171, team_id = 17,
                          athlete_id = 3149391)
```

---

`espn_wnba_game_probabilities`

# Get ESPN WNBA Event Win Probabilities

---

## Description

**Get ESPN WNBA Event Win Probabilities**

**Get ESPN WNBA Event Win Probabilities**

## Usage

```
espn_wnba_game_probabilities(event_id, limit = 200, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event/game identifier (character or numeric). |
| `limit` | integer. Maximum number of probability rows to return. Defaults to `200`. Pagination is handled internally. |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per play-level win-probability entry.

Columns as documented in the shared espn_basketball_game_probabilities_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_probabilities(event_id = "401736171", limit = 50)
```

---

`espn_wnba_game_propbets`

# Get ESPN WNBA Event Prop Bets (Long Format)

---

## Description

Returns prop-bet markets for one WNBA event + provider in long format: one row per (athlete × prop type). Each row has american / decimal / fraction odds plus the current target (e.g. over/under line). Hits the core-v2 `competitions/{id}/odds/{provider_id}/propBets` endpoint and auto-paginates.

## Usage

```
espn_wnba_game_propbets(event_id, provider_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| provider_id | Sportsbook provider id (e.g. 58 = ESPN BET, 100 = Caesars). Look up via espn_wnba_game_odds(). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Event Prop Bets (Long Format)**

## Value

A long tibble with one row per (athlete × prop type).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_propbets(event_id = 401736171, provider_id = 58)
```

espn_wnba_game_team_linescores
**Get ESPN WNBA Event Competitor Linescores (Per-Quarter)**

## Description

Returns the per-quarter scoring breakdown for one team in one NBA event. One row per period (regulation quarters + any overtime periods).

Returns the per-team statistical leaders for one WNBA event in long format: one row per (category x athlete rank). Categories typically include points, rebounds, assists, and rating.

Returns the game-day roster index for one team in one WNBA event. Each row carries the athlete id and the core-v2 `$ref` URL — use the ref to dereference athlete-game splits or biographical data.

Returns full team-game statistics for one team in one WNBA event in long format: one row per (category x stat). Covers offensive, defensive, and general categories with both raw values and display strings.

Returns team records as of the given WNBA event: overall, home, away, conference, and division breakdowns where available. One row per record type.

Returns a one-row tibble with one team's final score for one WNBA event: numeric `value`, display string, `winner` flag, and source metadata. Quick-lookup wrapper — use `espn_wnba_game_team_linescores()` for per-period detail.

## Usage

```
espn_wnba_game_team_linescores(event_id, team_id, ...)

espn_wnba_game_team_leaders(event_id, team_id, ...)

espn_wnba_game_team_roster(event_id, team_id, ...)

espn_wnba_game_team_statistics(event_id, team_id, ...)

espn_wnba_game_team_records(event_id, team_id, ...)

espn_wnba_game_team_score(event_id, team_id, ...)
```

## Arguments

| | |
|---|---|
| `event_id` | ESPN event identifier. |
| `team_id` | ESPN team identifier. |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per period.

A long tibble with one row per (category x rank).

A tibble with one row per active athlete.

A long tibble with one row per (category x stat).

A tibble with one row per record type.

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_team_linescores(event_id = 401736171, team_id = 17)


espn_wnba_game_team_leaders(event_id = 401736171, team_id = 17)


espn_wnba_game_team_roster(event_id = 401736171, team_id = 17)


espn_wnba_game_team_statistics(event_id = 401736171, team_id = 17)


espn_wnba_game_team_records(event_id = 401736171, team_id = 17)
```

```
espn_wnba_game_team_score(event_id = 401736171, team_id = 17)
```

---

espn_wnba_game_team_roster_entry **Get ESPN WNBA Event Competitor Roster Entry (Per-Athlete Game-Day Row)**

---

## Description

Returns a single-row tibble describing one athlete's game-day roster entry for one WNBA event. Carries the **starter** flag, **didNotPlay** flag with reason, ejection flag, and the substitution slot if the athlete came in for another player. Pair with espn_wnba_game_team_roster() to enumerate the roster.

## Usage

```
espn_wnba_game_team_roster_entry(event_id, team_id, athlete_id, ...)
```

## Arguments

| | |
|---|---|
| event_id | ESPN event identifier. |
| team_id | ESPN team identifier. |
| athlete_id | ESPN athlete identifier. |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(),

espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_game_team_roster_entry(event_id = 401736171,
                                 team_id = 17,
                                 athlete_id = 3149391)
```

---

# espn_wnba_injuries Get ESPN WNBA Injuries

---

## Description

**Get ESPN WNBA Injuries**

**Get ESPN WNBA Injuries**

## Usage

```
espn_wnba_injuries(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Numeric or character season year (e.g. 2025). The ESPN injury endpoint does not filter by season server-side; the value is attached as a constant column on the returned tibble for downstream joins. Defaults to most_recent_wnba_season(). |
| ... | Currently unused; reserved for future argument threading. |

## Value

Returns a tibble of league-wide WNBA injury records. Returns an empty tibble (zero rows) when no injuries are reported.

**Injuries**

Columns as documented in the shared espn_basketball_injuries_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_injuries()
```

---

`espn_wnba_leaders` **Get ESPN WNBA League Leaders**

---

## Description

**Get ESPN WNBA League Leaders**

**Get ESPN WNBA League Leaders**

## Usage

```
espn_wnba_leaders(season = most_recent_wnba_season(), season_type = 2, ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single wehoop_data tibble with one row per category-athlete pair.

Columns as documented in the shared espn_basketball_leaders_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_leaders(season = 2024, season_type = 2)
```

---

# espn_wnba_news Get ESPN WNBA News

---

## Description

**Get ESPN WNBA News**

**Get ESPN WNBA News**

## Usage

```
espn_wnba_news(limit = 50)
```

## Arguments

| | |
|---|---|
| limit | integer. Maximum number of articles to return. Default 50. |

## Details

Retrieve ESPN WNBA news. Uses getOption("wehoop.proxy") or http_proxy/https_proxy environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of news articles.

Columns as documented in the shared espn_basketball_news_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_news(limit = 5)
```

espn_wnba_player_awards

## Get ESPN WNBA Athlete Awards

### Description

**Get ESPN WNBA Athlete Awards**

**Get ESPN WNBA Athlete Awards**

### Usage

```
espn_wnba_player_awards(athlete_id, ...)
```

### Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

### Value

A single tibble. This endpoint is sparse; many athletes have no award data, in which case an empty tibble with canonical columns is returned.

Columns as documented in the shared espn_basketball_player_awards_schema table.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankin espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_player_awards(athlete_id = "3149391")
```

---

`espn_wnba_player_career_stats`

## Get ESPN WNBA Athlete Career Stats (Long Format)

---

## Description

Returns career stats for a WNBA athlete in long format. Default `stat_type = 0L` fetches the standard "All Splits" / regular-season view. Pass a vector like `c(0L, 1L, 2L)` to attempt multiple types and bind them via a `stat_type_id` column; variants that 404 for that athlete are silently skipped. Stat types: 0 = regular season (default endpoint), 1 = postseason, 2 = career aggregate. Coverage of types 1 and 2 is sparse — many athletes only have type 0 populated.

## Usage

```
espn_wnba_player_career_stats(athlete_id, stat_type = 0L, ...)
```

## Arguments

| | |
|---|---|
| `athlete_id` | ESPN athlete identifier. |
| `stat_type` | Integer or integer vector of stat-type codes. Default `0L` fetches the standard "All Splits" / regular-season view. Pass a vector like `c(0L, 1L, 2L)` to bind multiple types via a `stat_type_id` column; non-existent variants are silently skipped. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Athlete Career Stats (Long Format)**

## Value

A long tibble (one row per stat_type × split × category × stat).

Columns as documented in the shared espn_basketball_player_career_stats_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
# LeBron James — regular + postseason combined
espn_wnba_player_career_stats(athlete_id = 3149391)
# Just career aggregate
espn_wnba_player_career_stats(athlete_id = 3149391, stat_type = 2L)
```

---

`espn_wnba_player_eventlog_v2`

# Get ESPN WNBA Athlete Per-Season Event Log (core-v2)

---

## Description

Returns one row per (event x team) for an WNBA athlete's appearances in a given season. Distinct from espn_wnba_player_eventlog() which wraps the web-common-v3 gamelog endpoint returning stats per game; this core-v2 variant returns refs + `played` flag and is era-correct.

## Usage

```
espn_wnba_player_eventlog_v2(
  athlete_id,
  season = most_recent_wnba_season(),
  ...
)
```

### Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier. |
| season | Season year. Defaults to most recent WNBA season. |
| ... | Additional arguments; currently unused. |

### Value

A tibble with one row per event appearance. See package source for column schema.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

### Examples

```
espn_wnba_player_eventlog_v2(athlete_id = 2999102, season = 2025)
```

---

## espn_wnba_player_info Get ESPN WNBA Athlete Info

---

### Description

**Get ESPN WNBA Athlete Info**

**Get ESPN WNBA Athlete Info**

## Usage

```
espn_wnba_player_info(athlete_id, ...)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use options(wehoop.proxy = ...) – see ?wehoop for details. |

## Value

A named list of data frames: Bio, Team, Position, Status, College, Draft.

### Bio

Columns as documented in the shared espn_basketball_player_info_bio_schema table.

### Team

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| abbreviation | character | Short abbreviation. |
| display_name | character | Display name. |

### Position

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |
| abbreviation | character | Short abbreviation. |

### Status

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |
| type | character | Record type / category. |

### College

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| name | character | Display name. |
| mascot | character | Team mascot. |

**Draft**

| col_name | types | description |
|---|---|---|
| year | character | 4-digit year. |
| round | character | Tournament / playoff round. |
| selection | character | Selection. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_player_info(athlete_id = "3149391")
```

---

`espn_wnba_player_overview`

**Get ESPN WNBA Athlete Overview**

---

## Description

**Get ESPN WNBA Athlete Overview**

**Get ESPN WNBA Athlete Stats**

**Get ESPN WNBA Athlete Gamelog**

**Get ESPN WNBA Athlete Splits**

**Get ESPN WNBA Athlete Eventlog**

**Get ESPN WNBA Athlete Eventlog**

**Get ESPN WNBA Athlete Statisticslog**

## Usage

```
espn_wnba_player_overview(athlete_id, season = most_recent_wnba_season(), ...)

espn_wnba_player_stats_v3(athlete_id, season = most_recent_wnba_season(), ...)

espn_wnba_player_gamelog(athlete_id, season = most_recent_wnba_season(), ...)

espn_wnba_player_splits(athlete_id, season = most_recent_wnba_season(), ...)

espn_wnba_player_eventlog(athlete_id, season = most_recent_wnba_season(), ...)

espn_wnba_player_statisticslog(
  athlete_id,
  season = most_recent_wnba_season(),
  ...
)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Value

A named list of data frames: Statistics, NextGame, Last5Games, Headlines, FantasyOutlook.

**Statistics**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

**NextGame**

Columns as documented in the shared espn_basketball_player_overview_next_game_schema table.

**Last5Games**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

**Headlines**

| col_name | types | description |
|---|---|---|
| headline | character | News headline. |
| description | character | Long-form description text. |
| published | character | Publication timestamp (ISO 8601). |

**FantasyOutlook**

Columns as documented in the shared espn_basketball_player_overview_statistics_schema table.

A wide `wehoop_data` tibble, one row per athlete-season-team, with the ESPN stat categories spread across prefixed columns:

| col_name | types | description |
|---|---|---|
| athlete_id | character | ESPN athlete identifier (echoed input). |
| season | integer | Season year for the stat line. |
| team_id | character | ESPN team identifier for that season. |
| team_slug | character | Team slug (e.g. 'phoenix-mercury'). |
| avg_* | numeric | Per-game season-average stats (e.g. `avg_avg_points`). |
| tot_* | numeric | Season-total stats (e.g. `tot_points`). |
| misc_* | numeric | Miscellaneous season totals. |

Stat column names come from ESPN's positional `names` array per category, cleaned via `janitor::make_clean_names()`; the exact set varies by season.

A single tibble with one row per game. Column names reflect the stat labels returned by ESPN and will vary by season and player.

A single long-format tibble. When data are present, columns include at minimum `category` and `split_name`, plus per-stat columns driven by ESPN labels.

A single tibble. Per-event `statistics.$ref` URLs from the ESPN core-v2 API are returned as the character column `statistics_ref` and are NOT resolved. Similarly, `event_ref`, `competition_ref`, and `team_ref` are returned as character columns.

| col_name | types | description |
|---|---|---|
| event_ref | character | Reference link to the originating event. |
| competition_ref | character | Competition ref. |
| team_ref | character | Team ref. |
| statistics_ref | character | Statistics ref. |

A single tibble. When resolved, each row corresponds to one statistical entry in the core-v2 statistics log, with `event_ref` and `statistics_ref` character columns pointing to resolvable ESPN endpoints.

| col_name | types | description |
|---|---|---|
| event_ref | character | Reference link to the originating event. |
| statistics_ref | character | Statistics ref. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detai espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_player_overview(athlete_id = "3149391", season = 2024)

espn_wnba_player_stats_v3(athlete_id = "4068159", season = 2024)

espn_wnba_player_gamelog(athlete_id = "3149391", season = 2024)

espn_wnba_player_splits(athlete_id = "3149391", season = 2024)

espn_wnba_player_eventlog(athlete_id = "3149391", season = 2024)

espn_wnba_player_statisticslog(athlete_id = "3149391", season = 2024)
```

---

`espn_wnba_player_seasons`

# Get ESPN WNBA Athlete Career Seasons

---

## Description

Returns the list of seasons an WNBA athlete appeared in. Useful for bounding follow-up calls to per-season endpoints.

## Usage

```
espn_wnba_player_seasons(athlete_id, ...)
```

## Arguments

| | |
|---|---|
| athlete_id | ESPN athlete identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per career season.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| athlete_id | character | ESPN athlete id. |
| season | integer | Season year. |
| ref | character | $ref URL to the season detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankir espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week() espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
# LeBron James (1966): 23 career seasons
espn_wnba_player_seasons(athlete_id = 3149391)
```

espn_wnba_player_stats

**Get ESPN WNBA player stats data**

## Description

**Get ESPN WNBA player stats data**

## Usage

```
espn_wnba_player_stats(
  athlete_id,
  year,
  season_type = "regular",
  total = FALSE
)
```

## Arguments

| | |
|---|---|
| athlete_id | Athlete ID |
| year | Year |
| season_type | (character, default: regular): Season type - regular or postseason |
| total | (boolean, default: FALSE): Totals |

## Value

Returns a tibble with the player stats data

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(),

espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankin
espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week()
espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(),
espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail
espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(),
espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(),
espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_player_stats(athlete_id = 2529130, year = 2022))
```

---

# espn_wnba_position Get ESPN WNBA Position Detail

---

## Description

Returns metadata for a single WNBA position. Useful for dereferencing position `$ref` URLs embedded in athlete records, and for navigating parent/leaf relationships in the position taxonomy.

## Usage

```
espn_wnba_position(position_id, ...)
```

## Arguments

| | |
|---|---|
| position_id | ESPN position identifier (character or numeric). |
| ... | Additional arguments; currently unused. |

## Value

A single-row tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(),
espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(),
espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(),
espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(),
espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(),
espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(),
espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(),
espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(),

espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_position(position_id = 1)
```

---

`espn_wnba_positions` **Get ESPN WNBA Positions Index**

---

## Description

Returns the WNBA position dictionary index. One row per position with its id and the canonical `$ref` URL — pass an id to `espn_wnba_position()` for full details (display name, abbreviation, leaf flag, parent link).

## Usage

```
espn_wnba_positions(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per position.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_positions()
```

---

# espn_wnba_powerindex Get ESPN WNBA Season Power Index (Long Format)

---

## Description

Returns ESPN's Basketball Power Index (BPI) and related per-team metrics for one WNBA season, in long format: one row per (team x stat).

## Usage

```
espn_wnba_powerindex(
  season = most_recent_wnba_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| season_type | Integer (1=preseason, 2=regular (default), 3=postseason). |
| ... | Additional arguments; currently unused. |

## Value

A long tibble with one row per (team x stat).

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_powerindex(season = 2025)
```

---

`espn_wnba_scoreboard` **Get WNBA schedule for a specific year/date from ESPN's API**

---

## Description

**Get WNBA schedule for a specific year/date from ESPN's API**

## Usage

```
espn_wnba_scoreboard(season)
```

### Arguments

| | |
|---|---|
| season | Either numeric or character |

### Value

Returns a tibble

Columns as documented in the shared espn_basketball_scoreboard_schema table.

### Author(s)

Saiem Gilani.

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

### Examples

```
# Get schedule from date 2022-08-31

  try(espn_wnba_scoreboard (season = "20220831"))
```

---

## espn_wnba_season_awards

**Get ESPN WNBA Season Awards Index**

---

## Description

Returns the list of award IDs given out in a WNBA season from sports.core.api.espn.com/v2/sports/basketball/lea
The index only contains IDs and $ref URLs — pass an ID to espn_wnba_award() for the award name, description, and winners.

## Usage

```
espn_wnba_season_awards(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per award.

| col_name | types | description |
|---|---|---|
| season | integer | Season year. |
| award_id | character | ESPN award identifier. |
| ref | character | Full $ref URL for the award detail. |
| league | character | League slug ("wnba"). |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_awards(season = 2024)
```

---

espn_wnba_season_group

## Get ESPN WNBA Season Group Detail

---

## Description

Returns metadata for one group (conference or division) in one (WNBA season x season-type), plus `$ref` URLs to its parent group, children groups, member teams, and standings.

Returns the list of child groups (e.g. divisions within a conference) for one (WNBA season x season-type x parent-group).

Returns the list of team IDs that belong to one group (conference or division) for one (WNBA season x season-type).

## Usage

```
espn_wnba_season_group(
  group_id,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)

espn_wnba_season_group_children(
  group_id,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)

espn_wnba_season_group_teams(
  group_id,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| group_id | ESPN group identifier. |
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season Group Detail**

**Get ESPN WNBA Season Group Children Index**

**Get ESPN WNBA Season Group Teams Index**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_group_schema table.

A tibble with one row per child group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| parent_group_id | character | Parent group id (queried). |
| child_group_id | character | Child group id. |
| ref | character | `$ref` to child group detail. |

A tibble with one row per team in the group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| group_id | character | ESPN group id. |
| team_id | character | ESPN team id. |
| ref | character | `$ref` URL to the team-in-season entry. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`,

espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_group(group_id = 5, season = 2025)


espn_wnba_season_group_children(group_id = 5, season = 2025)


espn_wnba_season_group_teams(group_id = 5, season = 2025)
```

---

`espn_wnba_season_groups`

# Get ESPN WNBA Season Groups Index

---

## Description

Returns the list of group IDs (conferences / divisions) for one (WNBA season x season-type) via core-v2 `/seasons/{season}/types/{season_type}/groups`.

## Usage

```
espn_wnba_season_groups(
  season = most_recent_wnba_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| `season` | Season year. Defaults to most recent WNBA season. |
| `season_type` | Season-type id (2 = regular (default)). |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per group.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| group_id | character | ESPN group id. |
| ref | character | `$ref` URL for the group detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_info(), espn_wnba_season_leaders( espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_groups(season = 2025)
```

---

`espn_wnba_season_info` **Get ESPN WNBA Season Info**

---

## Description

**Get ESPN WNBA Season Info**

**Get ESPN WNBA Season Info**

## Usage

```
espn_wnba_season_info(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| season | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use options(wehoop.proxy = ...) – see ?wehoop for details. |

## Value

A named list of wehoop_data tibbles: Info, Types, Athletes, Coaches, Teams, Awards. $ref URL components are returned as character columns and are NOT auto-resolved – use targeted endpoint functions for details.

**Info**

| col_name | types | description |
|---|---|---|
| year | integer | 4-digit year. |
| start_date | character | Start date (YYYY-MM-DD). |
| end_date | character | End date (YYYY-MM-DD). |
| display_name | character | Display name. |
| type_id | character | Type identifier (numeric). |
| type_name | character | Type name. |

**Types / Athletes / Coaches / Teams / Awards**

| col_name | types | description |
|---|---|---|
| count | integer | Count of count. |
| ref | character | Ref. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(),

espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_leader espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_info(season = 2025)
```

---

espn_wnba_season_leaders

# Get ESPN WNBA Season Leaders (Long Format)

---

## Description

Returns the per-category leaderboard for one (WNBA season x season-type), in long format. Each row is one (category x rank) entry, e.g. "Points Per Game x rank 1 x LeBron James".

## Usage

```
espn_wnba_season_leaders(
  season = most_recent_wnba_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Season-type id (2 = regular (default), 3 = postseason). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season Leaders (Long Format)**

## Value

A long tibble with one row per (category x leader).

Columns as documented in the shared espn_basketball_season_leaders_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_leaders(season = 2025)
```

---

`espn_wnba_season_ranking`

# Get ESPN WNBA Season Ranking Detail

---

## Description

Returns the per-week snapshot index for one ranking source (e.g. AP Top 25). Each row is one weekly snapshot; the `ref` URL resolves to the actual ranked teams for that (season-type x week) and will be wrapped by a forthcoming `espn_LOWERWNBA_week_ranking()`.

## Usage

```
espn_wnba_season_ranking(ranking_id, season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| ranking_id | ESPN ranking identifier (character or numeric). |
| season | Season year. Defaults to most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season Ranking Detail**

## Value

A tibble with one row per weekly snapshot.

Columns as documented in the shared espn_basketball_season_ranking_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_ranking(ranking_id = 1, season = 2025)
```

`espn_wnba_season_rankings`

**Get ESPN WNBA Season Rankings Index**

## Description

Returns the index of season-level rankings recorded for one WNBA season. WNBA typically returns zero rankings (ranking polls are a college concept); the wrapper is provided for symmetry across leagues. For WBB use the matching `espn_wbb_season_rankings()`.

## Usage

```
espn_wnba_season_rankings(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year. Defaults to most recent WNBA season. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season Rankings Index**

## Value

A tibble with one row per ranking source.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| ranking_id | character | ESPN ranking id. |
| ref | character | `$ref` URL for the ranking detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`,

espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_rankings(season = 2025)
```

---

# espn_wnba_season_type Get ESPN WNBA Season-Type Detail

---

## Description

Returns metadata for one season-type within an WNBA season: name, abbreviation, start / end dates, and whether it carries groups, standings, or playoff legs.

## Usage

```
espn_wnba_season_type(
  season_type = 2L,
  season = most_recent_wnba_season(),
  ...
)
```

## Arguments

| | |
|---|---|
| season_type | Season-type id (1 = preseason, 2 = regular (default), 3 = postseason, 4 = off-season). |
| season | Season year. Defaults to most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season-Type Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_type_schema table.

### Author(s)

Saiem Gilani

### See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

### Examples

```
espn_wnba_season_type(season_type = 2, season = 2025)
```

---

`espn_wnba_season_types`

## Get ESPN WNBA Season Types Index

---

### Description

Returns the index of season-type IDs that exist for one WNBA season (typically 1 = preseason, 2 = regular, 3 = postseason, 4 = off-season). Pass an ID to espn_wnba_season_type() for the start/end dates and whether that type carries groups, standings, or legs.

### Usage

```
espn_wnba_season_types(season = most_recent_wnba_season(), ...)
```

### Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per season type.

| col_name | types | description |
|---|---|---|
| league | character | League slug ("wnba"). |
| season | integer | Season year. |
| season_type | integer | Season-type id (1/2/3/4). |
| ref | character | $ref URL for the type detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_types(season = 2025)
```

---

# espn_wnba_season_week Get ESPN WNBA Season-Week Detail

---

## Description

Returns metadata for one week (number, start / end dates, text label, and $ref to the per-week rankings endpoint).

## Usage

```
espn_wnba_season_week(
  week,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| week | Week number. |
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Season-Week Detail**

## Value

A single-row tibble.

Columns as documented in the shared espn_basketball_season_week_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_week(week = 5, season = 2025)
```

---

`espn_wnba_season_weeks`

**Get ESPN WNBA Season Weeks Index**

---

## Description

Returns the list of week IDs for one (WNBA season x season-type). WNBA uses a week structure inherited from ESPN's schema, but week-level rankings are populated only for college (MBB / WBB).

## Usage

```
espn_wnba_season_weeks(
  season = most_recent_wnba_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| `season` | Season year. Defaults to most recent WNBA season. |
| `season_type` | Season-type id (2 = regular (default), 3 = postseason). |
| `...` | Additional arguments; currently unused. |

## Value

A tibble with one row per week.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| week | integer | Week number (1-based). |
| ref | character | `$ref` URL for the week detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_season_weeks(season = 2025)
```

---

`espn_wnba_seasons` **Get ESPN WNBA Seasons**

---

## Description

**Get ESPN WNBA Seasons**

**Get ESPN WNBA Seasons**

## Usage

```
espn_wnba_seasons(...)
```

## Arguments

| | |
|---|---|
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single wehoop_data tibble with one row per season.

| col_name | types | description |
|---|---|---|
| season | integer | Season identifier (4-digit year or 'YYYY-YY' string). |
| start_date | character | Start date (YYYY-MM-DD). |
| end_date | character | End date (YYYY-MM-DD). |
| display_name | character | Display name. |
| season_type_count | integer | Count of season type. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_seasons()
```

---

# espn_wnba_standings Get ESPN WNBA Standings

---

## Description

**Get ESPN WNBA Standings**

## Usage

```
espn_wnba_standings(year)
```

## Arguments

| | |
|---|---|
| year | Either numeric or character (YYYY) |

## Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

## Author(s)

Geoff Hutchinson

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_standings(year = 2021))
```

`espn_wnba_team` **Get ESPN WNBA Team Detail**

## Description

Era-correct team identity for a WNBA franchise in a specific season, plus the available `$ref` URLs for deeper resources (record, statistics, leaders, coaches, transactions, etc.). Backed by the core-v2 endpoint `sports.core.api.espn.com/v2/sports/basketball/leagues/wnba/seasons/{season}/teams/{team_id}`.

Historical depth goes back to **1997** (WNBA founding). Older seasons return fewer `$ref` keys; missing refs become `NA`.

## Usage

```
espn_wnba_team(team_id, season = most_recent_wnba_season(), ...)

espn_wnba_team_roster(team_id, season = most_recent_wnba_season(), ...)

espn_wnba_team_leaders(team_id, season = most_recent_wnba_season(), ...)

espn_wnba_team_season_profile(team_id, season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier (character or numeric). |
| `season` | Season year (numeric). Defaults to the most recent WNBA season. |
| `...` | Additional arguments; currently unused. |

## Value

A named list of data frames: `Info`, `Record`, `NextEvent`, `StandingSummary`, `Coaches`.

### Info

Columns as documented in the shared espn_basketball_team_info_schema table.

### Record

| col_name | types | description |
|---|---|---|
| type | character | Record type / category. |
| summary | character | Summary. |
| stats | list | Stats. |

### NextEvent

Columns as documented in the shared espn_basketball_player_overview_next_game_schema table.

### StandingSummary

| col_name | types | description |
|---|---|---|
| standing_summary | character | Standing summary. |

**Coaches**

| col_name | types | description |
|---|---|---|
| id | character | Unique play identifcation number |
| first_name | character | Player's first name. |
| last_name | character | Player's last name. |
| experience | integer | Years of professional experience. |

A single tibble with one row per athlete.

Columns as documented in the shared espn_basketball_team_roster_schema table. ESPN's roster endpoint ignores the season, so a season before the current one is built from that season's published player box scores (name, jersey, position and headshot from the athlete's last game; bio columns `NA`), and errors when none are published.

A single long-format tibble (one row per category-rank-athlete).

Columns as documented in the shared espn_basketball_team_leaders_schema table.

A single-row tibble with team identity scalars and `_ref` URL columns. Selected columns:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`, `espn_wnba_position()`, `espn_wnba_positions()`, `espn_wnba_powerindex()`, `espn_wnba_scoreboard()`, `espn_wnba_season_awards()`, `espn_wnba_season_group()`, `espn_wnba_season_groups()`, `espn_wnba_season_info()`, `espn_wnba_season_leaders()`, `espn_wnba_season_ranking()`, `espn_wnba_season_rankings()`, `espn_wnba_season_type()`, `espn_wnba_season_types()`, `espn_wnba_season_week()`, `espn_wnba_season_weeks()`, `espn_wnba_seasons()`, `espn_wnba_standings()`, `espn_wnba_team_injuries()`, `espn_wnba_team_news()`, `espn_wnba_team_record()`, `espn_wnba_team_record_detail()`, `espn_wnba_team_schedule()`, `espn_wnba_team_season_roster()`, `espn_wnba_team_season_statistics()`, `espn_wnba_team_stats()`, `espn_wnba_teams()`, `espn_wnba_transactions()`, `espn_wnba_venues()`, `espn_wnba_week_ranking()`, `espn_wnba_week_rankings()`

## Examples

```
espn_wnba_team(team_id = "17", season = 2025)

espn_wnba_team_roster(team_id = "17", season = 2025)

espn_wnba_team_leaders(team_id = "17", season = 2025)

espn_wnba_team_season_profile(team_id = "17", season = 2025)
```

---

`espn_wnba_team_injuries`

# Get ESPN WNBA Team Injuries

---

## Description

**Get ESPN WNBA Team Injuries**

**Get ESPN WNBA Team Injuries**

## Usage

```
espn_wnba_team_injuries(team_id, ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier (character or numeric; passed as-is). Use `espn_wnba_teams()` to look up team IDs. Example: `"17"` (Las Vegas Aces). |
| `...` | Currently unused; reserved for future argument threading. |

## Value

Returns a tibble of injury records for the specified WNBA team. Returns an empty tibble (zero rows) when the team has no reported injuries.

**Injuries**

Columns as documented in the shared espn_basketball_team_injuries_injuries_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_injuries(team_id = "17")
```

---

# espn_wnba_team_news Get ESPN WNBA Team News

---

## Description

**Get ESPN WNBA Team News**

**Get ESPN WNBA Team News**

## Usage

```
espn_wnba_team_news(team_id, limit = 25)
```

## Arguments

| | |
|---|---|
| team_id | character or integer. ESPN team ID (e.g. 17 for Las Vegas Aces). |
| limit | integer. Maximum number of articles to return. Default 25. |

## Details

Retrieve ESPN WNBA news for a specific team. Uses `getOption("wehoop.proxy")` or `http_proxy`/`https_proxy` environment variables for proxy configuration (per-call proxy override is not supported for ESPN wrappers).

## Value

Returns a tibble of team news articles.

Columns as documented in the shared espn_basketball_news_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_news(team_id = "17", limit = 5)
```

---

# espn_wnba_team_record Get ESPN WNBA Team Record (Per Season Type)

---

## Description

Returns the long-format record breakdown for a WNBA team in one season and season-type. Each row is one record category.

## Usage

```
espn_wnba_team_record(
  team_id,
  season = most_recent_wnba_season(),
  season_type = c(2L, 3L),
  ...
)
```

## Arguments

| | |
|---|---|
| team_id | ESPN team identifier. |
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Integer (2 = regular season default). |
| ... | Additional arguments; currently unused. |

## Value

A tibble with one row per record category.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record_detail(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_record(team_id = 17, season = 2025)
```

---

espn_wnba_team_record_detail

# Get ESPN WNBA Team Record Detail (Long Format)

---

## Description

Returns one team's record detail in long format: one row per stat in the record's `stats[]` array. Use `espn_wnba_team_record()` to enumerate available `record_id` values per team-season (overall / home / away / conference + per-opponent breakdowns).

## Usage

```
espn_wnba_team_record_detail(team_id, season, record_id, season_type = 2L, ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year (numeric). |
| `record_id` | Record identifier (from `espn_wnba_team_record()` index). |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`,

espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_schedule(), espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_record_detail(team_id = 17, season = 2024, record_id = 0)
```

---

`espn_wnba_team_schedule`

# Get ESPN WNBA Team Schedule

---

## Description

**Get ESPN WNBA Team Schedule**

**Get ESPN WNBA Team Schedule**

## Usage

```
espn_wnba_team_schedule(
  team_id,
  season = most_recent_wnba_season(),
  season_type = 2,
  ...
)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier (character or numeric). |
| `season` | Season year (numeric). Defaults to the most recent WNBA season. |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused. |

## Value

A single tibble with one row per event.

Columns as documented in the shared espn_basketball_team_schedule_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_season_r, espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_schedule(team_id = "17", season = 2025)
```

---

`espn_wnba_team_season_roster`

# Get ESPN WNBA Team Roster (Per-Season, core-v2)

---

## Description

Returns the per-season team roster as a tibble of athlete IDs from `seasons/{y}/teams/{id}/athletes`. Distinct from `espn_wnba_team_roster()` which targets a site-v2 endpoint optimized for the current season; this core-v2 variant is era-correct and available back to ESPN's earliest season for each league.

## Usage

```
espn_wnba_team_season_roster(team_id, season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year. Defaults to most recent WNBA season. |
| `...` | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Team Roster (Per-Season, core-v2)**

## Value

A tibble with one row per athlete on the season roster.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| team_id | character | ESPN team id. |
| season | integer | Season year. |
| athlete_id | character | ESPN athlete id. |
| ref | character | `$ref` URL to athlete-in-season detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_season_roster(team_id = 17, season = 2025)
```

`espn_wnba_team_season_statistics` **Get ESPN WNBA Team Season Statistics (Long Format with Rank)**

## Description

Returns the full team-season-type statistics sheet for one WNBA team in long format: one row per (category x stat). Each row carries the team's league rank for that stat where ESPN provides it (`rank` + `rank_display_value`). Complements `espn_wnba_team_record()` (W-L only) with the full stat package.

## Usage

```
espn_wnba_team_season_statistics(
  team_id,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| `team_id` | ESPN team identifier. |
| `season` | Season year (numeric). Defaults to the most recent WNBA season. |
| `season_type` | Integer season type: 1 = preseason, 2 = regular (default), 3 = postseason. |
| `...` | Additional arguments; currently unused. |

## Value

A long tibble with one row per (category x stat).

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`,

espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_roster(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_team_season_statistics(team_id = 17, season = 2024)
```

---

# espn_wnba_team_stats Get ESPN WNBA team stats data

---

## Description

**Get ESPN WNBA team stats data**

## Usage

```
espn_wnba_team_stats(team_id, year, season_type = "regular", total = FALSE)
```

## Arguments

| | |
|---|---|
| team_id | Team ID |
| year | Year |
| season_type | (character, default: regular): Season type - regular or postseason |
| total | (boolean, default: FALSE): Totals |

## Value

Returns a tibble with the team stats data

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_team_stats(team_id = 17, year = 2020))
```

---

# espn_wnba_teams *Get ESPN's WNBA team names and ids*

---

## Description

Get ESPN's WNBA team names and ids

## Usage

```
espn_wnba_teams()
```

## Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking(), espn_wnba_week_rankings()

## Examples

```
try(espn_wnba_teams())
```

---

`espn_wnba_transactions`

# Get ESPN WNBA Transactions

---

## Description

**Get ESPN WNBA Transactions**

**Get ESPN WNBA Transactions**

## Usage

```
espn_wnba_transactions(season = most_recent_wnba_season(), limit = 100, ...)
```

## Arguments

| | |
|---|---|
| `season` | Season year (numeric, e.g. 2025). Defaults to the most recent WNBA season. |
| `limit` | Maximum number of transactions to return (integer). Default `100`. |
| `...` | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)`. |

## Details

Calls the ESPN site-v2 endpoint `https://site.api.espn.com/apis/site/v2/sports/basketball/wnba/transactions`. Releases have a `NULL` `to_team_id`; those are stored as `NA`. Returns an empty tibble rather than erroring when no transactions are available.

## Value

A `wehoop_data` tibble with one row per transaction:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`, `espn_wnba_position()`, `espn_wnba_positions()`, `espn_wnba_powerindex()`, `espn_wnba_scoreboard()`, `espn_wnba_season_awards()`, `espn_wnba_season_group()`, `espn_wnba_season_groups()`, `espn_wnba_season_info()`, `espn_wnba_season_leaders()`, `espn_wnba_season_ranking()`, `espn_wnba_season_rankings()`, `espn_wnba_season_type()`, `espn_wnba_season_types()`, `espn_wnba_season_week()`, `espn_wnba_season_weeks()`, `espn_wnba_seasons()`, `espn_wnba_standings()`, `espn_wnba_team()`, `espn_wnba_team_injuries()`, `espn_wnba_team_news()`, `espn_wnba_team_record()`, `espn_wnba_team_record_detail()`, `espn_wnba_team_schedule()`, `espn_wnba_team_season_roster()`, `espn_wnba_team_season_statistics()`, `espn_wnba_team_stats()`, `espn_wnba_teams()`, `espn_wnba_venues()`, `espn_wnba_week_ranking()`, `espn_wnba_week_rankings()`

## Examples

```
espn_wnba_transactions(season = 2025, limit = 10)
```

---

`espn_wnba_venues` **Get ESPN WNBA Venues**

---

## Description

**Get ESPN WNBA Venues**

**Get ESPN WNBA Venues**

## Usage

```
espn_wnba_venues(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments; currently unused but retained for forward compatibility. Proxy configuration should use `options(wehoop.proxy = ...)` – see `?wehoop` for details. |

## Value

A single `wehoop_data` tibble with one row per venue.

Columns as documented in the shared espn_basketball_venues_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: `espn_wnba_athletes_index()`, `espn_wnba_award()`, `espn_wnba_calendar()`, `espn_wnba_coach_season()`, `espn_wnba_coaches()`, `espn_wnba_conferences()`, `espn_wnba_draft()`, `espn_wnba_draft_athlete_detail()`, `espn_wnba_draft_pick()`, `espn_wnba_draft_rounds()`, `espn_wnba_franchise()`, `espn_wnba_franchises()`, `espn_wnba_freeagents()`, `espn_wnba_futures()`, `espn_wnba_game_all()`, `espn_wnba_game_odds()`, `espn_wnba_game_official_detail()`, `espn_wnba_game_play()`, `espn_wnba_game_play_personnel()`, `espn_wnba_game_player_box()`, `espn_wnba_game_probabilities()`, `espn_wnba_game_propbets()`, `espn_wnba_game_team_linescores()`, `espn_wnba_game_team_roster_entry()`, `espn_wnba_injuries()`, `espn_wnba_leaders()`, `espn_wnba_news()`, `espn_wnba_player_awards()`, `espn_wnba_player_career_stats()`, `espn_wnba_player_eventlog_v2()`, `espn_wnba_player_info()`, `espn_wnba_player_overview()`, `espn_wnba_player_seasons()`, `espn_wnba_player_stats()`, `espn_wnba_position()`, `espn_wnba_positions()`, `espn_wnba_powerindex()`, `espn_wnba_scoreboard()`, `espn_wnba_season_awards()`, `espn_wnba_season_group()`, `espn_wnba_season_groups()`, `espn_wnba_season_info()`, `espn_wnba_season_leaders()`, `espn_wnba_season_ranking()`, `espn_wnba_season_rankings()`, `espn_wnba_season_type()`, `espn_wnba_season_types()`, `espn_wnba_season_week()`, `espn_wnba_season_weeks()`, `espn_wnba_seasons()`, `espn_wnba_standings()`, `espn_wnba_team()`, `espn_wnba_team_injuries()`, `espn_wnba_team_news()`, `espn_wnba_team_record()`, `espn_wnba_team_record_detail()`, `espn_wnba_team_schedule`, `espn_wnba_team_season_roster()`, `espn_wnba_team_season_statistics()`, `espn_wnba_team_stats()`, `espn_wnba_teams()`, `espn_wnba_transactions()`, `espn_wnba_week_ranking()`, `espn_wnba_week_rankings()`

## Examples

```
espn_wnba_venues()
```

espn_wnba_week_ranking

**Get ESPN WNBA Per-Week Ranking Detail**

## Description

Returns the long-format ranked teams for one (season x season-type x week x ranking-source). WNBA typically returns an empty tibble.

## Usage

```
espn_wnba_week_ranking(
  ranking_id,
  week,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| ranking_id | Ranking source id (1 = AP, 2 = Coaches, etc.). |
| week | Week number. |
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Per-Week Ranking Detail**

## Value

A tibble with one row per ranked team (typically 25).

Columns as documented in the shared espn_basketball_week_ranking_schema table.

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_rankings()

## Examples

```
espn_wnba_week_ranking(ranking_id = 1, week = 5, season = 2025)
```

---

`espn_wnba_week_rankings`

# Get ESPN WNBA Per-Week Rankings Index

---

## Description

Returns the index of ranking sources available for one (WNBA season x season-type x week). WNBA does not publish weekly rankings, so this typically returns an empty tibble; the wrapper is provided for symmetry with college (MBB / WBB).

## Usage

```
espn_wnba_week_rankings(
  week,
  season = most_recent_wnba_season(),
  season_type = 2L,
  ...
)
```

## Arguments

| | |
|---|---|
| week | Week number. |
| season | Season year. Defaults to most recent WNBA season. |
| season_type | Season-type id (2 = regular (default)). |
| ... | Additional arguments; currently unused. |

## Details

**Get ESPN WNBA Per-Week Rankings Index**

## Value

A tibble with one row per ranking source.

| col_name | types | description |
|---|---|---|
| league | character | League slug. |
| season | integer | Season year. |
| season_type | integer | Season-type id. |
| week | integer | Week number. |
| ranking_id | character | ESPN ranking id. |
| ref | character | $ref URL for the ranked-teams detail. |

## Author(s)

Saiem Gilani

## See Also

Other ESPN WNBA Functions: espn_wnba_athletes_index(), espn_wnba_award(), espn_wnba_calendar(), espn_wnba_coach_season(), espn_wnba_coaches(), espn_wnba_conferences(), espn_wnba_draft(), espn_wnba_draft_athlete_detail(), espn_wnba_draft_pick(), espn_wnba_draft_rounds(), espn_wnba_franchise(), espn_wnba_franchises(), espn_wnba_freeagents(), espn_wnba_futures(), espn_wnba_game_all(), espn_wnba_game_odds(), espn_wnba_game_official_detail(), espn_wnba_game_play(), espn_wnba_game_play_personnel(), espn_wnba_game_player_box(), espn_wnba_game_probabilities(), espn_wnba_game_propbets(), espn_wnba_game_team_linescores(), espn_wnba_game_team_roster_entry(), espn_wnba_injuries(), espn_wnba_leaders(), espn_wnba_news(), espn_wnba_player_awards(), espn_wnba_player_career_stats(), espn_wnba_player_eventlog_v2(), espn_wnba_player_info(), espn_wnba_player_overview(), espn_wnba_player_seasons(), espn_wnba_player_stats(), espn_wnba_position(), espn_wnba_positions(), espn_wnba_powerindex(), espn_wnba_scoreboard(), espn_wnba_season_awards(), espn_wnba_season_group(), espn_wnba_season_groups(), espn_wnba_season_info(), espn_wnba_season_leaders(), espn_wnba_season_ranking(), espn_wnba_season_rankings(), espn_wnba_season_type(), espn_wnba_season_types(), espn_wnba_season_week(), espn_wnba_season_weeks(), espn_wnba_seasons(), espn_wnba_standings(), espn_wnba_team(), espn_wnba_team_injuries(), espn_wnba_team_news(), espn_wnba_team_record(), espn_wnba_team_record_detail(), espn_wnba_team_schedule, espn_wnba_team_season_roster(), espn_wnba_team_season_statistics(), espn_wnba_team_stats(), espn_wnba_teams(), espn_wnba_transactions(), espn_wnba_venues(), espn_wnba_week_ranking()

## Examples

```
espn_wnba_week_rankings(week = 5, season = 2025)
```

---

`feb_game_info` **FEB Game Info**

---

## Description

**Game header of one FEB game: competition, tip-off, venue, officials, status, both teams and the score.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/TeamStats/{game_id}` (every game route carries the header; this reads the lightest one).

## Usage

```
feb_game_info(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request`

(HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Part |
| comp_id | integer | Competition id (constant across seasons): 115 Primera FEB (LEB Oro), 118 Segunda FEB, 2 |
| competition | character | Competition name as printed that season, uppercase ('PRIMERA FEB'; 2020-21 'LIGA LEB |
| group_label | character | The game's group (grupo) label, trimmed: 'Unico' = the single group, '"A"' = group A. It is t |
| period | integer | Current or last period number; 4 for a finished regulation game. |
| clock | character | 'FINAL' once the game is over, else the game clock 'mm:ss' remaining in the period. |
| start_time | POSIXct | Tip-off in Spanish local time (POSIXct, tz Europe/Madrid): the source carries no offset, so th |
| venue | character | Venue (pabellon) name. |
| address | character | Venue street address, free text. |
| referee1 | character | Crew chief (arbitro principal), 'SURNAMES, NAME'. |
| referee2 | character | First umpire (arbitro auxiliar 1), 'SURNAMES, NAME'. |
| referee3 | character | Second umpire (arbitro auxiliar 2), 'SURNAMES, NAME'; null when the game had two refe |
| status | integer | Game state code: 3 = finished (status_text FINISHED). Null on 2020-21 games. |
| status_text | character | Game state label in English ('FINISHED'). Null on 2020-21 games. |
| home_team_id | integer | Home team-season id (key space: FEB team registrations, new every season). |
| home_team | character | Home team name for the season (sponsor name, uppercase). |
| home_team_code | character | Home federation team code, 8 digits as a string. Null on 2020-21 games. |
| home_club_code | character | Home federation club code, 6 digits as a string. Null on 2020-21 games. |
| home_score | integer | Home points: the final score when clock is FINAL, else the live score. |
| away_team_id | integer | Away team-season id (key space: FEB team registrations, new every season). |
| away_team | character | Away team name for the season (sponsor name, uppercase). |
| away_team_code | character | Away federation team code, 8 digits as a string. Null on 2020-21 games. |
| away_club_code | character | Away federation club code, 6 digits as a string. Null on 2020-21 games. |
| away_score | integer | Away points: the final score when clock is FINAL, else the live score. |

## See Also

Other FEB Functions: `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_period_scores()`, `feb_player_box()`, `feb_results()`, `feb_schedule()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

## Examples

```
try(feb_game_info(game_id = 2535725))
```

`feb_leaders` **FEB Game Leaders**

## Description

**Each team's top players of one FEB game by points, rebounds, assists and fouls.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/Ranking/{game_id}`

One row per team, category and ranked player (`rank` 1 = the leader, in the order the feed lists them).

## Usage

```
feb_leaders(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

```
feb_leaders(game_id = 2535725)
```

## Value

A wehoop_data tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido. |
| team_side | character | Side of the game the row belongs to: home or away (the feed's TEAM[0] / TEAM[1], or its 1/2 ar |
| team_id | integer | Team-season id (key space: FEB team registrations; a NEW id every season, e.g. HLA Alicante = |
| team | character | Team name as registered for the season (sponsor name, uppercase: 'HLA ALICANTE'). |
| category | character | Leader category: points, rebounds, assists or fouls (the feed's Puntos, Rebotes, Asistencias, Faltas |
| rank | integer | Place in the team's category list, 1 = the leader, in the order the feed lists them. |
| player_id | integer | FEB person id: stable across seasons and clubs (A. Jorda Rodriguez = 1676991 in 2025-26 and 20 |
| shirt_number | character | Shirt number (dorsal) as a string, so '00' and '0' stay distinct. The shot feed names the shooter by |
| player | character | Player name as 'I. SURNAME(S)', uppercase ('V. AGUILAR HERNANDEZ'). |
| starter | logical | True when the player was in the starting five (the page's I column), false when on the bench. |
| captain | logical | True for the team captain. Null on 2020-21 games, where the feed does not send it. |
| value | integer | The player's count in the row's category (points, rebounds, assists or personal fouls). |

## See Also

Other FEB Functions: feb_game_info(), feb_overview(), feb_pbp(), feb_period_scores(), feb_player_box(), feb_results(), feb_schedule(), feb_shots(), feb_standings(), feb_team_stats()

---

# feb_overview FEB Overview

---

## Description

**Today's (or the latest) games of every live FEB competition, with scores and status.**

Endpoint: GET https://intrafeb.feb.es/LiveStats.API/api/v1/overview

One row per game across the competitions with games in the window. An off day with no competitions raises wehoop_no_data; a game listed under another competition is refused (wehoop_asset_fetch).

## Usage

```
feb_overview()
```

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's sportsdataverse.basketball.feb wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt

answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

```
feb_overview()
```

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| comp_id | integer | Competition id (constant across seasons): 115 Primera FEB (LEB Oro), 118 Segunda FEB, |
| comp_code | integer | Season-specific competition edition code (e.g. 23473 = Primera FEB 2026-27). |
| competition | character | Competition name as printed that season, uppercase ('PRIMERA FEB'; 2020-21 'LIGA LE |
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route. |
| round_id | integer | Round id; 0 in every sample (the route does not fill it). |
| stage | character | Stage code as a string; '0' in every sample (meaning unverified). |
| venue | character | Name of the venue (pabellon) where the game is played. |
| start_time | POSIXct | Tip-off in Spanish local time (POSIXct, tz Europe/Madrid; the source has no offset). |
| start_time_utc | POSIXct | Tip-off in UTC (POSIXct, tz UTC). |
| period | integer | Current or last period (4 when finished). |
| clock | character | 'FINAL' once finished, else the game clock ('00:00' before tip-off). |
| status | integer | Game state code: 3 = finished, 1 = scheduled (seen 2026-10-12); the live codes are unverifie |
| home_team_id | integer | Home team-season id (new every season; joins the schedule pages). |
| home_team | character | Home team name for the season. |
| home_team_code | character | Home federation team code, 8 digits as a string. |
| home_club_code | character | Home federation club code, 6 digits as a string. |
| home_abbreviation | character | Home team three-letter abbreviation ('PAL'). |
| home_score | integer | Home points (0 before tip-off). |
| away_team_id | integer | Away team-season id (new every season; joins the schedule pages). |
| away_team | character | Away team name for the season. |
| away_team_code | character | Away federation team code, 8 digits as a string. |

| | | |
|---|---|---|
| away_club_code | character | Away federation club code, 6 digits as a string. |
| away_abbreviation | character | Away team three-letter abbreviation. |
| away_score | integer | Away points (0 before tip-off). |
| period_scores | character | Points per period as 'home-away' pairs joined by ';' ('16-16;19-16;10-21;14-14'); null befo |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_pbp()`, `feb_period_scores()`, `feb_player_box()`, `feb_results()`, `feb_schedule()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

---

`feb_pbp` **FEB Play-by-Play**

---

## Description

**Play-by-play of one FEB game (KeyFacts): every shot, rebound, foul, substitution and time-out, in Spanish text with coded actions, oldest play first.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/KeyFacts/{game_id}`

A body whose plays name another game is refused (`wehoop_asset_fetch`), not returned. 2020-21 games are text only: `action`, the ids, the parameters, the position and the log times are `NA` there.

## Usage

```
feb_pbp(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido |
| play_num | integer | Play sequence number, 1 = the first play of the game; unique per game. The parser sorts by it (th |
| period | integer | Period number of the play. |
| clock | character | Game clock REMAINING in the period as 'mm:ss' (it counts down from 10:00). |
| clock_seconds | integer | The clock as whole seconds remaining in the period (08:34 = 514), derived by the parser. |
| team_side | character | Side of the acting team: home (feed code 1) or away (2); null on period lines. |
| team_id | integer | Team-season id of the acting team (new every season); null on period lines and 2020-21 games. |
| player_id | integer | FEB person id of the actor (stable across seasons); null on team, period and 2020-21 lines. |
| license_id | integer | Per-season player licence id of the actor (changes when the player re-registers; A. Jorda Rodrigu |
| action | character | Action code: shoot, fthrow, rebound, assist, recovery (steal), lose (turnover), blockshot, foul, sub |
| text | character | Play description in Spanish: '(TEAM) I. SURNAME: ACTION (running count)'; 2020-21 game |
| score_home | integer | Home score AFTER the play; filled on scoring plays only. |
| score_away | integer | Away score AFTER the play; filled on scoring plays only. |
| x | double | Court x position of a shoot line: percent (0-100) of the length of a horizontal full-court image, or |
| y | double | Court y position of a shoot line: percent (0-100) of the width from the top edge (the same scale a |
| log_param1 | character | Action parameter 1 (string). Team and player actions: a pseudo-GUID '00000000-0000-000{sid |
| log_param2 | character | Action parameter 2 (string). The actor's licence id on shoot, fthrow, foul, rebound, assist, recover |
| log_param3 | character | Action parameter 3 (string). fthrow: '1' made, '0' missed; subst: the outgoing player's licence; f |
| log_param4 | character | Action parameter 4 (string). shoot and fthrow: '1' made, '0' missed; foul: the licence of the foul |
| log_param5 | character | Action parameter 5 (string); action-dependent, meaning unverified. |
| log_param6 | character | Action parameter 6 (string). shoot: shot value '2' or '3'; fthrow: '1'; foul: free throws awarded. |
| log_param7 | character | Action parameter 7 (string); action-dependent, meaning unverified. |
| log_param8 | character | Action parameter 8 (string): a small integer on shoot, foul and lose lines (by distribution a zone c |
| log_param9 | character | Action parameter 9 (string); blank or null in every sampled game. |
| log_param10 | character | Action parameter 10 (string): '1' on almost every foul; meaning unverified. |
| log_time | POSIXct | Wall-clock time the play was logged, Spanish local time (POSIXct, tz Europe/Madrid). NA on 2 |
| log_time_utc | POSIXct | The log time in UTC (POSIXct, tz UTC). NA on 2020-21 games. |

## See Also

Other FEB Functions: feb_game_info(), feb_leaders(), feb_overview(), feb_period_scores(), feb_player_box(), feb_results(), feb_schedule(), feb_shots(), feb_standings(), feb_team_stats()

## Examples

```
try(feb_pbp(game_id = 2535725))
```

---

`feb_period_scores` **FEB Period Scores**

---

## Description

**Points each team scored in each period of one FEB game.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/TeamStats/{game_id}`

One row per period played; the points are scored IN the period, not cumulative. Overtime numbering is unverified.

## Usage

```
feb_period_scores(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of feb_schedule(), feb_results() and feb_overview(). |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

```
feb_period_scores(game_id = 2535725)
```

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido.asp |
| period | integer | Period number (1-4 regulation; overtime numbering is unverified, no overtime game is in the sample |
| home_points | integer | Points the home team scored IN this period (not cumulative). |
| away_points | integer | Points the away team scored IN this period (not cumulative). |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_player_box()`, `feb_results()`, `feb_schedule()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

---

`feb_player_box` **FEB Player Box Score**

---

## Description

**Per-player box score of one FEB game: minutes, points, shooting splits, rebounds, assists, fouls and plus/minus.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/BoxScore/{game_id}`

One row per player and team. `p1` / `p2` / `p3` are free throws, two- and three-point field goals (`m` made, `a` attempted, `p` percentage 0-100); `seconds_played` is the provider's `min` (seconds, not minutes); `player_id` is the FEB person id (stable across seasons) and `shirt_number` stays a string (`"00"` is not `"0"`).

## Usage

```
feb_player_box(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partic |
| team_side | character | Side of the game the row belongs to: home or away (the feed's TEAM[0] / TEAM[1], or its 1/2 |
| team_id | integer | Team-season id (key space: FEB team registrations; a NEW id every season, e.g. HLA Alicante |
| team | character | Team name as registered for the season (sponsor name, uppercase: 'HLA ALICANTE'). |
| player_id | integer | FEB person id: stable across seasons and clubs (A. Jorda Rodriguez = 1676991 in 2025-26 and |
| shirt_number | character | Shirt number (dorsal) as a string, so '00' and '0' stay distinct. The shot feed names the shooter |
| player | character | Player name as 'I. SURNAME(S)', uppercase ('V. AGUILAR HERNANDEZ'). |
| starter | logical | True when the player was in the starting five (the page's I column), false when on the bench. |
| captain | logical | True for the team captain. Null on 2020-21 games, where the feed does not send it. |
| seconds_played | integer | Seconds on court (the feed's min, an integer count of seconds: 1064 = 17:44; a team row sums i |
| min_formatted | character | Time played as 'mm:ss'. |
| pts | integer | Points scored (the team's points on team rows; the final score when the clock is FINAL, else liv |
| p1m | integer | Free throws made (tiros libres; integer as a string). |
| p1a | integer | Free throws attempted (integer as a string). |
| p1p | double | Free-throw percentage, 0-100, as a string with a DECIMAL COMMA ('77,3'); '0,0' when no a |

| | | |
|---|---|---|
| p2m | integer | Two-point field goals made (integer as a string). |
| p2a | integer | Two-point field goals attempted (integer as a string). |
| p2p | double | Two-point percentage, 0-100, string with a decimal comma ('54,1'). |
| p3m | integer | Three-point field goals made (integer as a string). |
| p3a | integer | Three-point field goals attempted (integer as a string). |
| p3p | double | Three-point percentage, 0-100, string with a decimal comma ('47,6'). |
| fgm | integer | Field goals made, 2PT + 3PT (tiros de campo; integer as a string). |
| fga | integer | Field goals attempted, 2PT + 3PT (integer as a string). |
| fgp | double | Field-goal percentage, 0-100, string with a decimal comma ('51,7'). |
| ro | integer | Offensive rebounds (rebotes ofensivos; integer as a string). |
| rd | integer | Defensive rebounds (rebotes defensivos; integer as a string). |
| rt | integer | Total rebounds (rebotes totales; integer as a string). |
| reb | integer | Total rebounds (the same count as rt). |
| assist | integer | Assists (asistencias; integer as a string). |
| st | integer | Steals (balones robados, page column BR; integer as a string). |
| to | integer | Turnovers (balones perdidos, page column BP; integer as a string). |
| bs | integer | Blocks made (tapones a favor, page column TF; integer as a string). |
| tc | integer | Shots of this player/team that were blocked (tapones en contra / recibidos, page column TC; int |
| mt | integer | Dunks (mates, page column MT; integer as a string). |
| pf | integer | Personal fouls committed (faltas cometidas, page column FC; integer as a string). |
| rf | integer | Fouls received / drawn (faltas recibidas, page column FR; integer as a string). |
| pllss | integer | Plus/minus while the player was on court (balance, page column +/-), signed integer. |
| val | integer | FEB efficiency rating (valoracion): points + rebounds + assists + steals + blocks + fouls drawn |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_period_scores()`, `feb_results()`, `feb_schedule()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

## Examples

```
try(feb_player_box(game_id = 2535725))
```

---

`feb_results` **FEB Results**

---

## Description

**Latest round's results and the next round's fixtures of a FEB competition season.**

Endpoint: `GET https://baloncestoenvivo.feb.es/resultados.aspx?g={group}&t={season - 1}`

`section` is `results` (the LATEST round of the shown phase; for a finished season the last playoff round) or `next_round` (empty once the season is over). `comp_id` ties the site group to the API's competition id (group 1 -> 115, 2 -> 118, 4 -> 210).

## Usage

```
feb_results(group = 4, season)
```

## Arguments

| | |
|---|---|
| group | (*integer* default 4, Liga Femenina Endesa): The site's menu group: 1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa, 9 and 10 LF-2, 67 LF Challenge, 73 Copa de Espana, 74 Liga U. A group is not a competition id (see comp_id). |
| season | (*integer* required): Season END year (2027 = 2026-27), sent to the site as its START-year key t = season - 1. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's sportsdataverse.basketball.feb wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (intrafeb.feb.es) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of baloncestoenvivo.feb.es at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (season = 1997).

**Seasons** are END years (season = 2027 is 2026-27); the site's own t key is the START year, so the request sends t = season - 1.

**Errors** are classed conditions, never a warning plus an empty frame: wehoop_no_data (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), wehoop_bad_request (HTTP 400 / 422, or a game_id, group or season that is not a non-negative integer) and wehoop_asset_fetch (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit wehoop_error.

## Value

A wehoop_data tibble with these columns:

| col_name | type | description |
|---|---|---|
| group | integer | The site's menu group (1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa); not a compet |
| season | integer | Season END year (2027 = 2026-27), the site's START-year key plus one. |
| competition | character | Competition name on the page header ('PRIMERA FEB'). |

| | | |
|---|---|---|
| comp_id | integer | Competition id (CompID) read from the page; ties the site group to the API (group 1 -> 115, 2 -> |
| section | character | Which table the row comes from: results (the latest round) or next_round (the following fixtures) |
| round_title | character | Title of the table as printed ('Resultados Jornada 3(09/10/2026)'). |
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido |
| home_team_id | integer | Home team-season id (new every season; joins feb_overview and the game wrappers). |
| home_team | character | Home team name, uppercase. |
| away_team_id | integer | Away team-season id (new every season). |
| away_team | character | Away team name, uppercase. |
| result_text | character | Result cell verbatim: 'home-away' points when played; before the game the calendar shows the |
| home_score | integer | Home points; null before the game. |
| away_score | integer | Away points; null before the game. |
| status | character | played when the result cell holds a score, scheduled otherwise. |
| date | Date | Game date; null for played games on the schedule page (use round_date). |
| time | character | Tip-off 'HH:MM', Spanish local time; null for played games on the schedule page. |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_period_scores()`, `feb_player_box()`, `feb_schedule()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

## Examples

```
try(feb_results(group = 1, season = 2027))
```

---

`feb_schedule` **FEB Schedule**

---

## Description

**Full schedule of one group (phase) of a FEB competition season: every round, game, result and team-season id.**

Endpoint: `GET https://baloncestoenvivo.feb.es/calendario.aspx?g={group}&t={season - 1}`

Played games carry no date or time on this page: use `round_date` or `feb_game_info()`'s `start_time`. The page renders only the FIRST phase of the group (Segunda FEB shows group A); other phases and the playoffs need an ASP.NET postback and are not modelled. A page for another group or season is refused (`wehoop_asset_fetch`), an unknown group is `wehoop_no_data`.

## Usage

```
feb_schedule(group = 4, season)
```

## Arguments

| | |
|---|---|
| group | (*integer* default 4, Liga Femenina Endesa): The site's menu group: 1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa, 9 and 10 LF-2, 67 LF Challenge, 73 Copa de Espana, 74 Liga U. A group is not a competition id (see comp_id). |
| season | (*integer* required): Season END year (2027 = 2026-27), sent to the site as its START-year key t = season - 1. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's sportsdataverse.basketball.feb wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (intrafeb.feb.es) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of baloncestoenvivo.feb.es at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (season = 1997).

**Seasons** are END years (season = 2027 is 2026-27); the site's own t key is the START year, so the request sends t = season - 1.

**Errors** are classed conditions, never a warning plus an empty frame: wehoop_no_data (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), wehoop_bad_request (HTTP 400 / 422, or a game_id, group or season that is not a non-negative integer) and wehoop_asset_fetch (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit wehoop_error.

## Value

A wehoop_data tibble with these columns:

| col_name | type | description |
|---|---|---|
| group | integer | The site's menu group (1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa); not a compet |
| season | integer | Season END year (2027 = 2026-27), the site's START-year key plus one. |
| competition | character | Competition name on the page header ('PRIMERA FEB'). |
| round | integer | Round number (jornada) the game belongs to. |
| round_label | character | Round heading as printed ('Jornada 4 16/10/2026'). |
| round_date | Date | Date of the round (the heading's date); played games carry no date of their own on this page. |

| | | |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido |
| home_team_id | integer | Home team-season id (new every season; joins feb_overview and the game wrappers). |
| home_team | character | Home team name, uppercase. |
| away_team_id | integer | Away team-season id (new every season). |
| away_team | character | Away team name, uppercase. |
| result_text | character | Result cell verbatim: 'home-away' points when played; before the game the calendar shows the |
| home_score | integer | Home points; null before the game. |
| away_score | integer | Away points; null before the game. |
| status | character | played when the result cell holds a score, scheduled otherwise. |
| weekday | character | English weekday of a scheduled game (translated from the Spanish cell); null for played games. |
| date | Date | Game date; null for played games on the schedule page (use round_date). |
| time | character | Tip-off 'HH:MM', Spanish local time; null for played games on the schedule page. |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_period_scores()`, `feb_player_box()`, `feb_results()`, `feb_shots()`, `feb_standings()`, `feb_team_stats()`

## Examples

```
try(feb_schedule(group = 1, season = 2027))
```

---

`feb_shots` **FEB Shots**

---

## Description

**Every field-goal attempt of one FEB game with its court position (percent of a full-court image).**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/ShotChart/{game_id}`

Free throws are not in the feed. `x` / `y` are percent (0-100) of a horizontal full-court image, origin top-left; teams change ends at half-time. The feed names the shooter by shirt number: `player_id` is resolved through the game's roster (side + shirt number) and is `NA` when that pair is not unique.

## Usage

```
feb_shots(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

```
feb_shots(game_id = 2535725)
```

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partido |
| period | integer | Period number of the attempt. |
| clock | character | Game clock REMAINING in the period at the shot, 'mm:ss' (the play-by-play line's time). |
| clock_seconds | integer | The clock as whole seconds remaining in the period (08:34 = 514), derived by the parser. |
| team_side | character | Side of the shooter: home (feed code 0) or away (feed code 1; play-by-play uses 1/2). |
| team_id | integer | Team-season id of the shooter's team (from the shot chart's TEAM list); new every season. |
| shirt_number | character | Shirt number (dorsal) as a string, so '00' and '0' stay distinct. The shot feed names the shooter b |
| player_id | integer | FEB person id of the shooter, resolved from side + shirt number in the game's roster; null when t |
| made | logical | True when the shot was made, false when missed. Free throws are not in the feed. |
| x | double | Court x position: percent (0-100) of the length of a horizontal full-court image, origin top-left. T |
| y | double | Court y position: percent (0-100) of the width of a horizontal full-court image from the top edge |

## See Also

Other FEB Functions: feb_game_info(), feb_leaders(), feb_overview(), feb_pbp(), feb_period_scores(), feb_player_box(), feb_results(), feb_schedule(), feb_standings(), feb_team_stats()

---

`feb_standings` **FEB Standings**

---

## Description

**League table of a FEB competition season (live standings, or the final classification once it is over).**

Endpoint: `GET https://baloncestoenvivo.feb.es/resultados.aspx?g={group}&t={season - 1}`

`table` is `live` (the in-season table) or `final` (the final classification: only `position` and the team, every other statistic is `NA`); the page shows one of the two. Only the first phase of the group is shown.

## Usage

```
feb_standings(group = 4, season)
```

## Arguments

| | |
|---|---|
| `group` | (*integer* default `4`, Liga Femenina Endesa): The site's menu group: 1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa, 9 and 10 LF-2, 67 LF Challenge, 73 Copa de Espana, 74 Liga U. A group is not a competition id (see `comp_id`). |
| `season` | (*integer* required): Season END year (`2027` = 2026-27), sent to the site as its START-year key `t = season - 1`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| group | integer | The site's menu group (1 Primera FEB, 2 Segunda FEB, 4 Liga Femenina Endesa); not a compet |
| season | integer | Season END year (2027 = 2026-27), the site's START-year key plus one. |
| competition | character | Competition name on the page header ('PRIMERA FEB'). |
| comp_id | integer | Competition id (CompID) read from the page; ties the site group to the API (group 1 -> 115, 2 -> |
| table | character | live = the in-season league table; final = the final classification once the season is over. |
| position | integer | Rank in the table (Po). |
| team_id | integer | Team-season id (new every season; joins the schedule and game wrappers). |
| team | character | Team name as printed in the table (sponsor name, uppercase). |
| played | integer | Games played (PJ); null in the final classification. |
| won | integer | Games won (PG); null in the final classification. |
| lost | integer | Games lost (PP); null in the final classification. |
| points_for | integer | Points scored (PF); null in the final classification. |
| points_against | integer | Points conceded (PC); null in the final classification. |
| point_diff | integer | Point difference (DIF), signed; null in the final classification. |
| win_pct | double | Win percentage, 0-100 (the page's '100,0%', decimal comma converted). |
| streak | integer | Current streak as a signed integer: +3 = three wins in a row, -2 = two losses. |
| streak_text | character | The streak cell verbatim ('+3'). |

## See Also

Other FEB Functions: `feb_game_info()`, `feb_leaders()`, `feb_overview()`, `feb_pbp()`, `feb_period_scores()`, `feb_player_box()`, `feb_results()`, `feb_schedule()`, `feb_shots()`, `feb_team_stats()`

## Examples

```
try(feb_standings(group = 1, season = 2027))
```

`feb_team_stats` **FEB Team Stats**

## Description

**Team totals of both teams in one FEB game: shooting, rebounds, assists, turnovers, fouls.**

Endpoint: `GET https://intrafeb.feb.es/LiveStats.API/api/v1/TeamStats/{game_id}`

One row per team, home first. `fastb`, `ptsto`, `secpts`, `bench` and `paint` are null on every sampled game (the stat crews do not fill them); `seconds_played` and `val` are null on these rows.

## Usage

```
feb_team_stats(game_id)
```

## Arguments

| | |
|---|---|
| `game_id` | (*integer* required): FEB game id, e.g. `2535725`; the `game_id` column of `feb_schedule()`, `feb_results()` and `feb_overview()`. |

## Details

The Spanish Basketball Federation (FEB) wrappers cover Primera FEB, Segunda FEB and Liga Femenina Endesa (plus the federation's other groups on the same routes). They mirror sdv-py's `sportsdataverse.basketball.feb` wrappers of the same names: same arguments, the same snake_case columns and types, one row per sdv-py row.

**Legal notice.** The FEB's aviso legal (legal notice) asserts intellectual-property rights over the site's content and its databases (the EU database right). It has no explicit scraping clause, and robots.txt answers 404 on both hosts. Fetching for your own use on your own machine is how these wrappers are meant to be used; check the notice before you republish the data.

**Unofficial and unsupported.** The LiveStats API (`intrafeb.feb.es`) is the federation's own widget backend, not a documented public API. It needs a bearer token, which is public: it is read from the home page of `baloncestoenvivo.feb.es` at run time, kept in memory for the R session (24 h validity, renewed once on an HTTP 401) and never printed, logged or written. Requests are spaced at least one second apart per host; 429 and 5xx answers are retried (three attempts).

**Coverage.** The game routes answer from 2020-21 (Liga Femenina Endesa from its first game; Primera FEB from 2021-03-12, so its earlier 2020-21 games and every earlier season answer 404). 2020-21 play-by-play is text only. The schedule pages go back to 1996-97 (`season = 1997`).

**Seasons** are END years (`season = 2027` is 2026-27); the site's own `t` key is the START year, so the request sends `t = season - 1`.

**Errors** are classed conditions, never a warning plus an empty frame: `wehoop_no_data` (HTTP 404, an unknown group, a schedule page with no games, an empty overview window), `wehoop_bad_request` (HTTP 400 / 422, or a `game_id`, `group` or `season` that is not a non-negative integer) and `wehoop_asset_fetch` (any other failure: another status, a transport error, a token the API still rejects after one renewal, a non-JSON or non-HTML body, an error or changed envelope, a play-by-play of another game, a page for another group or season, or layout drift). All inherit `wehoop_error`.

```
feb_team_stats(game_id = 2535725)
```

## Value

A `wehoop_data` tibble with these columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FEB game id (one global sequence): the {gameId} of every LiveStats route and the p= of Partic |
| team_side | character | Side of the game the row belongs to: home or away (the feed's TEAM[0] / TEAM[1], or its 1/2 |
| team_id | integer | Team-season id (key space: FEB team registrations; a NEW id every season, e.g. HLA Alicante |
| team | character | Team name as registered for the season (sponsor name, uppercase: 'HLA ALICANTE'). |
| team_code | character | Federation team code, 8 digits as a string: the 6-digit club_code plus a 2-digit team suffix ('030 |
| club_code | character | Federation club code, 6 digits as a string (the first 6 digits of team_code). Whether it is stable ac |
| seconds_played | integer | Seconds on court (the feed's min, an integer count of seconds: 1064 = 17:44; a team row sums i |
| min_formatted | character | Time played as 'mm:ss'; '0:00' on /TeamStats rows (only box-score totals carry a value). |
| pts | integer | Team points: the final score when the game is over, else the live score. |
| p1m | integer | Free throws made (tiros libres; integer as a string). |
| p1a | integer | Free throws attempted (integer as a string). |
| p1p | double | Free-throw percentage, 0-100, as a string with a DECIMAL COMMA ('77,3'); '0,0' when no a |
| p2m | integer | Two-point field goals made (integer as a string). |
| p2a | integer | Two-point field goals attempted (integer as a string). |
| p2p | double | Two-point percentage, 0-100, string with a decimal comma ('54,1'). |
| p3m | integer | Three-point field goals made (integer as a string). |
| p3a | integer | Three-point field goals attempted (integer as a string). |
| p3p | double | Three-point percentage, 0-100, string with a decimal comma ('47,6'). |
| fgm | integer | Field goals made, 2PT + 3PT (tiros de campo; integer as a string). |
| fga | integer | Field goals attempted, 2PT + 3PT (integer as a string). |
| fgp | double | Field-goal percentage, 0-100, string with a decimal comma ('51,7'). |
| ro | integer | Offensive rebounds (rebotes ofensivos; integer as a string). |
| rd | integer | Defensive rebounds (rebotes defensivos; integer as a string). |
| rt | integer | Total rebounds (rebotes totales; integer as a string). |
| assist | integer | Assists (asistencias; integer as a string). |
| st | integer | Steals (balones robados, page column BR; integer as a string). |
| to | integer | Turnovers (balones perdidos, page column BP; integer as a string). |
| bs | integer | Blocks made (tapones a favor, page column TF; integer as a string). |
| tc | integer | Shots of this team that were blocked (tapones en contra, page column TC). Null on /TeamStats |
| mt | integer | Dunks (mates, page column MT). Null on /TeamStats rows. |
| pf | integer | Personal fouls committed (faltas cometidas, page column FC; integer as a string). |
| rf | integer | Fouls received / drawn (faltas recibidas, page column FR; integer as a string). |
| fouls | integer | Team fouls committed in the game (equals pf on the samples). |
| val | integer | FEB efficiency rating (valoracion). Null on /TeamStats rows, filled on box-score totals. |
| fastb | integer | Fast-break points. Null on every sampled game: the federation's stat crews do not fill it. |
| ptsto | integer | Points off turnovers. Null on every sampled game: the federation's stat crews do not fill it. |
| secpts | integer | Second-chance points. Null on every sampled game: the federation's stat crews do not fill it. |
| bench | integer | Bench points. Null on every sampled game: the federation's stat crews do not fill it. |
| paint | integer | Points in the paint. Null on every sampled game: the federation's stat crews do not fill it. |

## See Also

Other FEB Functions: feb_game_info(), feb_leaders(), feb_overview(), feb_pbp(), feb_period_scores(), feb_player_box(), feb_results(), feb_schedule(), feb_shots(), feb_standings()

---

# fiba_ecw_competition FIBA competition: EuroCup Women (FIBA Women's European Club Competitions Tier 2)

---

## Description

Per-competition aliases of the FIBA wrappers for EuroCup Women (FIBA Women's European Club Competitions Tier 2) (GDAP competition family 165), named as in sdv-py. Season rule: END year (2024-25 = 2025).

## Usage

```
fiba_ecw_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_ecw_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_ecw_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (END year (2024-25 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_ecw_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_ecw_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See `fiba_gdap_awards()` for the shared notes.

A `comp` of another competition family, or one GDAP gives no family, is `wehoop_asset_fetch`; a season with no edition is `wehoop_no_data`; a season listing several editions, or a game function called with neither `season` nor `comp`, is `wehoop_bad_request`.

```
fiba_ecw_editions(season = 2025)
fiba_ecw_schedule(season = 2025)
```

## Value

A `wehoop_data` tibble with the columns of the base wrapper (the raw JSON list when `return_parsed = FALSE`), one table per function:

`fiba_ecw_editions()`:

The columns of `fiba_gdap_editions()`:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |

| | | |
|---|---|---|
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_ecw_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is know |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |

| | | |
|---|---|---|
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gan |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_ecw_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a natio |

`fiba_ecw_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |

| | | |
|---|---|---|
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_ecw_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |

| | | |
|---|---|---|
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_ecw_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |

| | | |
|---|---|---|
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_ecw_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not rec |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |

| | | |
|---|---|---|
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_ecw_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: `fiba_elw_competition`, `fiba_wafrobasket_competition`, `fiba_wamericup_competition`, `fiba_wasiacup_competition`, `fiba_weurobasket_competition`, `fiba_woly_competition`, `fiba_wwc_competition`

---

`fiba_elw_competition` **FIBA competition: EuroLeague Women (FIBA Women's European Club Competitions Tier 1)**

---

## Description

Per-competition aliases of the FIBA wrappers for EuroLeague Women (FIBA Women's European Club Competitions Tier 1) (GDAP competition family 211), named as in sdv-py. Season rule: END year (2024-25 = 2025).

## Usage

```
fiba_elw_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_elw_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_elw_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (END year (2024-25 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_elw_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_elw_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See fiba_gdap_awards() for the shared notes.

A comp of another competition family, or one GDAP gives no family, is wehoop_asset_fetch; a season with no edition is wehoop_no_data; a season listing several editions, or a game function called with neither season nor comp, is wehoop_bad_request.

```
fiba_elw_editions(season = 2025)
fiba_elw_schedule(season = 2025)
```

## Value

A wehoop_data tibble with the columns of the base wrapper (the raw JSON list when return_parsed = FALSE), one table per function:

`fiba_elw_editions()`:

The columns of `fiba_gdap_editions()`:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_elw_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |

| | | |
|---|---|---|
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is know |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gam |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |

| | | |
|---|---|---|
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_elw_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a natio |

`fiba_elw_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |

`fiba_elw_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |

| | | |
|---|---|---|
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_elw_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |

| | | |
|---|---|---|
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_elw_pbp()`:

The columns of fiba_gamepage_pbp():

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not re |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

fiba_elw_shots():

The columns of fiba_gamepage_shots():

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |

| | | |
|---|---|---|
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: fiba_ecw_competition, fiba_wafrobasket_competition, fiba_wamericup_competition, fiba_wasiacup_competition, fiba_weurobasket_competition, fiba_woly_competition, fiba_wwc_competition

---

`fiba_gamepage_box_players`

# FIBA GAMEPAGE: box players

---

## Description

Player box score of one FIBA game.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_box_players(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `game_id` | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list (the game-page objects `game`, `playByPlay`, `gameDetails`, `playersTeamA`, `playersTeamB`). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises wehoop_no_data; a game still live raises wehoop_asset_fetch. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than comp, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-p |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |

| | | |
|---|---|---|
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_ranking()`, `fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_sta`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

## Examples

```
try(fiba_gamepage_box_players(game_id = 100720, comp = 208182))
```

fiba_gamepage_box_teams

**FIBA GAMEPAGE: box teams**

## Description

Team box score of one FIBA game, with the game-level figures.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_box_teams(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `game_id` | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list (the game-page objects `game`, `playByPlay`, `gameDetails`, `playersTeamA`, `playersTeamB`). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises `wehoop_no_data`; a game still live raises `wehoop_asset_fetch`. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than `comp`, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |

| | | |
|---|---|---|
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_ranking()`, `fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_sta`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

## Examples

```
try(fiba_gamepage_box_teams(game_id = 100720, comp = 208182))
```

`fiba_gamepage_lineup` **FIBA GAMEPAGE: lineup**

## Description

Lineup card (the players who were registered for the game) of both teams of one FIBA game.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_lineup(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `game_id` | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list (the game-page objects `game`, `playByPlay`, `gameDetails`, `playersTeamA`, `playersTeamB`). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's `T_<id>` and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises `wehoop_no_data`; a game still live raises `wehoop_asset_fetch`. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than `comp`, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gamepage_lineup(game_id = 100720, comp = 208182)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-page |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say). |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany); |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play p |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_ranking()`, `fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_sta`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

fiba_gamepage_pbp **FIBA GAMEPAGE: pbp**

## Description

Play-by-play of one FIBA game: every action with period, clock, running score, actor and shot x/y.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_pbp(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list (the game-page objects game, playByPlay, gameDetails, playersTeamA, playersTeamB). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises `wehoop_no_data`; a game still live raises `wehoop_asset_fetch`. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than `comp`, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not re |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gamepage_pbp(game_id = 100720, comp = 208182))
```

---

`fiba_gamepage_period_scores`

# FIBA GAMEPAGE: period scores

---

## Description

Points by period of both teams of one FIBA game.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_period_scores(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `game_id` | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list (the game-page objects `game`, `playByPlay`, `gameDetails`, `playersTeamA`, `playersTeamB`). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises `wehoop_no_data`; a game still live raises `wehoop_asset_fetch`. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than `comp`, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game- |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germ |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| points | integer | Points the team scored in the period. |

| | | |
|---|---|---|
| points_first_five_min | integer | Points the team scored in the first five minutes of the period (matches the play-by-play exc |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gamepage_period_scores(game_id = 100720, comp = 208182))
```

---

# fiba_gamepage_shots **FIBA GAMEPAGE: shots**

---

## Description

Shot attempts of one FIBA game (two- and three-pointers and free throws) with their x/y.

Endpoint: `GET https://www.fiba.basketball/en/games/{game_id}`

## Usage

```
fiba_gamepage_shots(game_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | (integer, required) FIBA game id (global): gameId of a gdap schedule row, e.g. 100720 = the World Cup 2023 final. |
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). The game must belong to it. Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list (the game-page objects game, playByPlay, gameDetails, playersTeamA, playersTeamB). |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

The wrapper takes the game id and its competition (comp) and builds the real page URL from GDAP (the competition's marketing name gives the history slug, its schedule gives the two team codes; a competition with an event page is tried at its event URL when the archive URL answers the not-found shell, and the Champions League host last).

Box scores start about 2006; play-by-play with shot x/y about 2012 (OVR events) and 2017-18 (FLS club competitions). A game with no box score raises `wehoop_no_data`; a game still live raises `wehoop_asset_fetch`. The page request sends a browser User-Agent (sdv-py answers fiba.basketball's HTTP 403 to plain clients with Chrome TLS; libcurl with a browser User-Agent was served on 2026-10-18). Shot x/y are image pixels with an unverified scale.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id, a game page with no box score, a listed game whose teams are not drawn yet); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season, a game of another competition than `comp`, the HTTP 200 not-found shell of a wrong game page, a live game). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |

| | | |
|---|---|---|
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gamepage_shots(game_id = 100720, comp = 208182))
```

---

`fiba_gdap_awards` **FIBA GDAP: awards**

---

## Description

Awards of a competition (MVP, All-Star Five...).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionawardsbyid`

## Usage

```
fiba_gdap_awards(comp, discipline_format = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| discipline_format | (character, optional) Optional; the site passes its discipline format. 5x5 answers the same as omitting it (verified on awards and rosters) |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_awards(comp = 208182)
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = 208182 |
| competition_end_date | character | Last day of the competition, YYYY-MM-DD. |
| award_date | character | Date awarded, YYYY-MM-DD. |
| competition_season | integer | Season key, three rules: the END year for club competitions (BCL 2024-25 = 2025) |
| competition_official_name | character | Official name of the competition edition the award was given in. |
| award_name | character | Award name (All-Star 5, MVP...). |

| | | |
|---|---|---|
| award_code | character | Award code (ALLSTAR, MVP...). |
| award_level_id | integer | Id of the object the award is attached to: the competitionId when awardLevel is Com |
| award_level | character | Level the award is given at: Competition. |
| award_label | character | Extra label; null in every sample. |
| player_id | integer | FIBA person id, global across competitions and roles (player, coach, official): roster |
| player_first_name | character | Awarded player's first name. |
| player_last_name | character | Awarded player's last name. |
| player_date_of_birth | character | Awarded player's date of birth, YYYY-MM-DD. |
| player_nationality | character | Awarded player's nationality, FIBA code. |
| teams | character | Teams the awarded player played for in the competition. Stored as a JSON string. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

# `fiba_gdap_competition` FIBA GDAP: competition

---

## Description

One competition (edition): season, dates, hosts, zone, category and competition family.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getGdapCompetitionById`

## Usage

```
fiba_gdap_competition(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |

| | | |
|---|---|---|
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_competition(comp = 208182))
```

---

# fiba_gdap_editions FIBA GDAP: editions

---

## Description

Every edition of a competition family between two seasons.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionsbyseasonsandcompetitionmark

## Usage

```
fiba_gdap_editions(
  family_id,
  season = NULL,
  to_season = NULL,
  return_parsed = TRUE
)
```

## Arguments

| | |
|---|---|
| family_id | (integer, required) Required. Competition family id (competitionMarketingName.competitionMarketingNameId): 201 = World Cup, 320 = men's Olympics, 306 = Women's World Cup, 112 = BCL tier 1, 211 = EuroLeague Women. |
| season | (integer, optional) Season key: the END year for club competitions (BCL 2024-25 = 2025), the event year for national-team events (World Cup 2023 = 2023); qualifiers carry the TARGET tournament year (World Cup 2027 qualifiers played in 2025 = 2027). Sent unchanged. First season, inclusive; omitted = no lower bound. |
| to_season | (integer, optional) Last season, inclusive (same key as season). Omitted with season given = the same season. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |

| | | |
|---|---|---|
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_federation_games(), fiba_gdap_federation_ranking(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta, fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_editions(family_id = 201, season = 2023))
```

`fiba_gdap_federation_games`

# FIBA GDAP: federation games

## Description

A federation's recent and upcoming national-team games.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapfederationgamesbyfederationid`

## Usage

```
fiba_gdap_federation_games(federation_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `federation_id` | (integer, required) Required. Federation organisationId, stable across competitions (64 = Germany; teamA.organisationId) |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_federation_games(federation_id = 64)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across competitions (GDAP key |
| game_name | character | Technical game name '<roundId>-<groupPairingCo |
| game_number | character | Game number within the round or pairing (a string: '1' |
| status_code | character | Game status code: VALID (validated result), CLOS (cl |
| team_a_team_id | integer | Competition team id: one per team per competition (G |
| team_a_organisation_id | integer | Organisation id, stable across competitions and season |
| team_a_code | character | Team code (national: FIBA country code; club: a club |
| team_a_official_name | character | Official name of team A: the country for a national tea |
| team_a_short_name | character | Short name of team A: the country code for a national |
| team_b_team_id | integer | Competition team id: one per team per competition (G |
| team_b_organisation_id | integer | Organisation id, stable across competitions and season |
| team_b_code | character | Team code (national: FIBA country code; club: a club |
| team_b_official_name | character | Official name of team B: the country for a national tea |
| team_b_short_name | character | Short name of team B: the country code for a national |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being played. |
| current_period | character | Last period reached ('Q4', 'OT1'). |
| chrono | character | Game clock of the current period, 'mm:ss' ('00:00' wh |
| live_game_status | integer | Live state code: 999 on every finished game; 5 seen on |
| current_period_status | character | State of the current period: E = ended (the only value s |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FIBA code in getgdapga |
| host_country_code | character | FIBA code of the venue's country. |
| venue_id | integer | Venue id (GDAP key space; 0 = unknown). |
| venue_name | character | Name of the arena the game is played in. |
| game_date_time | character | Tip-off in venue-local wall-clock time, ISO 8601 witho |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 without the Z. |
| has_time_game_date_time | logical | False when only the date is known (the time part is the |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in hours (Float64: a half-h |
| is_postponed | logical | True when the game was postponed. |
| is_played_behind_closed_doors | logical | True when played without spectators. |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as recorded by FIBA. |
| statistic_system | character | Live-stats system that scored the game: OVR (FIBA's |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairing number (knockout |
| round_round_id | integer | Round id (GDAP key space, unique across competition |
| round_round_code | character | Round code (GP group phase, QF, SF, F final, RS regul |
| round_round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| round_round_number | integer | Round number within the competition (1-based). |
| round_round_type | character | Round type label: Group Round, Non Group Round, S |
| round_round_status_code | character | Round status: COMPL completed, PROGR in progress |

| | | |
|---|---|---|
| round_game_system | character | Pairing system label (Single, Home and Away, Best of |
| round_game_system_code | character | Pairing system code: S single game, HA home and awa |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59:59.9999999 on the l |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key space): one per editio |
| competition_competition_code | character | Competition code (WMM men's World Cup, WOLYM |
| competition_season | integer | Season key, three rules: the END year for club competi |
| competition_official_name | character | Edition title (FIBA Basketball World Cup; Paris 2024 M |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, Completed, Closed. |
| competition_age_category | character | Age category (Senior, Under 16...). |
| competition_gender | character | Men or Women: the gender of the competition. |
| competition_fiba_zone | character | FIBA zone the competition belongs to (World, Europe, |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone table key |
| competition_zone_information_zone_official_name | character | Official name of the competition's FIBA zone (FIBA E |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, values u |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in the samples, values ur |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club competitions. |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, OLY Olympic Tourna |
| competition_competition_category_name | character | Name of the competition's category (FIBA Basketball |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a separate platform). |
| competition_competition_marketing_name | character | Competition family, shared by every edition; always nu |
| game_statistic_status_code | character | Box-score validation status: VALID. |
| game_play_by_play_status_code | character | Play-by-play availability flag: VALID or EMPTY. UNI |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, overtime remark); null wh |
| window_code | character | Qualifier window code; null otherwise. |
| window_name | character | Qualifier window name; null otherwise. |
| team_a_from | character | How team A qualified for this slot ('Winner of Game 8 |
| team_b_from | character | How team B qualified for this slot ('Winner of Game 8 |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_ranking()`, `fiba_gdap_final_standing` `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`,

fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_federation_ranking`

## FIBA GDAP: federation ranking

---

### Description

FIBA world ranking of national teams.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapfederationsranking`

### Usage

```
fiba_gdap_federation_ranking(
  category = "women",
  zone_id = 1L,
  as_of_date = NULL,
  return_parsed = TRUE
)
```

### Arguments

| | |
|---|---|
| `category` | (character, default `"women"` in wehoop; sdv-py and hoopR default `"men"`) Ranking category: men or women (the site also knows boys/girls) |
| `zone_id` | (integer, default `1L`) Ranking zone id from getgdaprankingzones (1 = World, 2 = Africa, 3 = Americas, 4 = Asia, 5 = Europe) |
| `as_of_date` | (character, optional) Ranking as of this date, ISO 8601: the latest ranking published on or before it (2026-03-10 answers the 2026-03-03 ranking). Omitted = the latest ranking |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

### Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_federation_ranking(category = "women", zone_id = 1, as_of_date = "2025-09-15")
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| as_of_date | character | Date of the ranking returned: the latest ranking published on or before the requeste |
| zone_ranking_code | character | Ranking zone code (WORLD, AFRICA...). |
| zone_ranking_official_name | character | Ranking zone official name. |
| world_rank | integer | Position in the FIBA world ranking (1 = best). |
| country_name | character | Country name in English. |
| zone_rank | integer | Rank within the zone. |
| ioc_code | character | IOC country code. |
| fiba_code | character | FIBA three-letter country code (GER, USA; historic URS, YUG). |
| current_points | double | Ranking points (one decimal). |
| world_rank_variation | integer | Change in world rank since the previous release (positive = up). |
| comments | character | Ranking note; null in every sample. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

fiba_gdap_final_standings

## FIBA GDAP: final standings

### Description

Final ranking of a competition.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionfinalstandingbycompetitionid

### Usage

```
fiba_gdap_final_standings(comp, return_parsed = TRUE)
```

### Arguments

| | |
|---|---|
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

### Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_final_standings(comp = 208182))
```

---

# fiba_gdap_game FIBA GDAP: game

---

## Description

One game's header: teams, score, venue, attendance and status codes (no box score or play-by-play: see fiba_gamepage).

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapgamebyid

## Usage

```
fiba_gdap_game(game_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | (integer, required) Required. FIBA game id (global GDAP key space): gameId of a schedule row; the <gameId> of a game page URL |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across competitions (GDAP key space): gdapGameI |
| round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| technical_name | character | Technical game name '<roundId>-<groupPairingCode>-<gameNumber>' |
| round_type | character | Round type label: Group Round, Non Group Round, Super Round (getgda |
| group_pairing_code | character | Group letter (group play) or pairing number (knockout bracket slot). |
| game_number | character | Game number within the round or pairing (a string: '1', 'A'). |
| status_code | character | Game status code: VALID (validated result), CLOS (closed, pre-2000 rows |
| game_day | integer | Game day number within the round. |
| team_a_from | character | How team A qualified for this slot ('Winner of Game 87', '1st of group C' |
| team_b_from | character | How team B qualified for this slot ('Winner of Game 88', '2nd of group A' |
| team_a_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at |
| team_a_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federa |
| team_a_code | character | Team code (national: FIBA country code; club: a club abbreviation such as |

| | | |
|---|---|---|
| team_a_official_name | character | Official name of team A: the country for a national team, the club for a clu |
| team_a_short_name | character | Short name of team A: the country code for a national team, the club's sho |
| team_b_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at |
| team_b_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federa |
| team_b_code | character | Team code (national: FIBA country code; club: a club abbreviation such as |
| team_b_official_name | character | Official name of team B: the country for a national team, the club for a clu |
| team_b_short_name | character | Short name of team B: the country code for a national team, the club's sho |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| team_a_aggregated_score | integer | Two-leg aggregate score of team A (0 when not a two-leg tie). |
| team_b_aggregated_score | integer | Two-leg aggregate score of team B (0 when not a two-leg tie). |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FIBA code in getgdapgameswithleaderdetai |
| host_country_code | character | FIBA code of the venue's country. |
| venue_id | integer | Venue id (GDAP key space; 0 = unknown). |
| venue_name | character | Name of the arena the game is played in. |
| game_date_time | character | Tip-off in venue-local wall-clock time, ISO 8601 without offset (T00:00:00 |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 without the Z. |
| has_time_game_date_time | logical | False when only the date is known (the time part is then 00:00:00). |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in hours (Float64: a half-hour zone such as In |
| is_postponed | logical | True when the game was postponed. |
| is_played_behind_closed_doors | logical | True when played without spectators. |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as recorded by FIBA. |
| is_live | logical | True while the game is being played. |
| current_period | character | Last period reached ('Q4', 'OT1'). |
| chrono | character | Game clock of the current period, 'mm:ss' ('00:00' when finished). |
| live_game_status | integer | Live state code: 999 on every finished game; 5 seen on a finished 2024 you |
| current_period_status | character | State of the current period: E = ended (the only value sampled). |
| game_statistic_status_code | character | Box-score validation status: VALID. |
| game_play_by_play_status_code | character | Play-by-play availability flag: VALID or EMPTY. UNRELIABLE: World |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, overtime remark); null when none. |
| window_id | integer | Qualifier window id; null otherwise. |
| window_code | character | Qualifier window code; null otherwise. |
| window_name | character | Qualifier window name; null otherwise. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank` `fiba_gdap_final_standings()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()` `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_sta` `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`,

fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_game(game_id = 100720))
```

---

`fiba_gdap_game_officials`

# FIBA GDAP: game officials

---

## Description

Officials nominated to one game (referees, table officials, commissioner).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapgamebyid`

## Usage

```
fiba_gdap_game_officials(game_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `game_id` | (integer, required) Required. FIBA game id (global GDAP key space): gameId of a schedule row; the <gameId> of a game page URL |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer

that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_game_officials(game_id = 100720)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64) of the game the official was nominated to. |
| person_id | integer | FIBA person id, global across competitions and roles (player, coach, official): roster personI |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnominationsbyid). |
| role_code | character | Role code (REF, FIBATDEL...). |
| role_name | character | Role name on the game (Crew chief, Umpire, Commissioner, Assistant scorer...). |
| role_order | character | Order within the role; a numeric string or null. |
| default_nationality | character | FIBA code of the official's main nationality. |
| license_nationality | character | FIBA code of the federation that licenses the official. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_stats()`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

---

`fiba_gdap_game_periods`

# FIBA GDAP: game periods

---

## Description

Period scores of one game.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapgamebyid`

## Usage

```
fiba_gdap_game_periods(game_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | (integer, required) Required. FIBA game id (global GDAP key space): gameId of a schedule row; the <gameId> of a game page URL |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64) of the game the period belongs to. |
| name | character | Period name ('Q1'...'OT1'). |
| number | integer | Period number (1-based). |
| team_a_score | integer | Team A points in the period. |
| team_b_score | integer | Team B points in the period. |
| team_a_from | character | How team A qualified for this slot ('Winner of Game 87', '1st of group C'); null in group play. |
| team_b_from | character | How team B qualified for this slot ('Winner of Game 88', '2nd of group A'); null in group play. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank: fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_game_periods(game_id = 100720))
```

---

`fiba_gdap_games_between`

# FIBA GDAP: games between

---

## Description

Games of EVERY FIBA competition in a local-time window (how competitions and games are discovered).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapgamesbetweentwodates`

## Usage

```
fiba_gdap_games_between(date_from, date_to, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `date_from` | (character, required) Required. Window start as YYYY-MM-DD, compared with the venue-local gameDateTime (not UTC). |
| `date_to` | (character, required) Required. Window end as YYYY-MM-DD, EXCLUSIVE (date_from = date_to returns nothing). |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full

intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_games_between(date_from = "2023-09-08", date_to = "2023-09-11")
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across competitions (GDAP key |
| game_name | character | Technical game name '<roundId>-<groupPairingCod |
| game_number | character | Game number within the round or pairing (a string: '1' |
| status_code | character | Game status code: VALID (validated result), CLOS (cl |
| team_a_team_id | integer | Competition team id: one per team per competition (G |
| team_a_organisation_id | integer | Organisation id, stable across competitions and seasons |
| team_a_code | character | Team code (national: FIBA country code; club: a club |
| team_a_official_name | character | Official name of team A: the country for a national tea |
| team_a_short_name | character | Short name of team A: the country code for a national |
| team_b_team_id | integer | Competition team id: one per team per competition (G |
| team_b_organisation_id | integer | Organisation id, stable across competitions and seasons |
| team_b_code | character | Team code (national: FIBA country code; club: a club |
| team_b_official_name | character | Official name of team B: the country for a national tea |
| team_b_short_name | character | Short name of team B: the country code for a national |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being played. |
| current_period | character | Last period reached ('Q4', 'OT1'). |
| chrono | character | Game clock of the current period, 'mm:ss' ('00:00' wh |
| live_game_status | integer | Live state code: 999 on every finished game; 5 seen on |
| current_period_status | character | State of the current period: E = ended (the only value s |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FIBA code in getgdapga |

| | | |
|---|---|---|
| host_country_code | character | FIBA code of the venue's country. |
| venue_id | integer | Venue id (GDAP key space; 0 = unknown). |
| venue_name | character | Name of the arena the game is played in. |
| game_date_time | character | Tip-off in venue-local wall-clock time, ISO 8601 witho |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 without the Z. |
| has_time_game_date_time | logical | False when only the date is known (the time part is the |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in hours (Float64: a half-h |
| is_postponed | logical | True when the game was postponed. |
| is_played_behind_closed_doors | logical | True when played without spectators. |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as recorded by FIBA. |
| statistic_system | character | Live-stats system that scored the game: OVR (FIBA's C |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairing number (knockout |
| round_round_id | integer | Round id (GDAP key space, unique across competition |
| round_round_code | character | Round code (GP group phase, QF, SF, F final, RS regul |
| round_round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| round_round_number | integer | Round number within the competition (1-based). |
| round_round_type | character | Round type label: Group Round, Non Group Round, S |
| round_round_status_code | character | Round status: COMPL completed, PROGR in progress |
| round_game_system | character | Pairing system label (Single, Home and Away, Best of |
| round_game_system_code | character | Pairing system code: S single game, HA home and awa |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59:59.9999999 on the l |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key space): one per editio |
| competition_competition_code | character | Competition code (WMM men's World Cup, WOLYM |
| competition_season | integer | Season key, three rules: the END year for club competi |
| competition_official_name | character | Edition title (FIBA Basketball World Cup; Paris 2024 M |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, Completed, Closed. |
| competition_age_category | character | Age category (Senior, Under 16...). |
| competition_gender | character | Men or Women: the gender of the competition. |
| competition_fiba_zone | character | FIBA zone the competition belongs to (World, Europe, |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone table key |
| competition_zone_information_zone_official_name | character | Official name of the competition's FIBA zone (FIBA E |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, values u |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in the samples, values un |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club competitions. |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, OLY Olympic Tourna |
| competition_competition_category_name | character | Name of the competition's category (FIBA Basketball |

| | | |
|---|---|---|
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a separate platform). |
| competition_competition_marketing_name | character | Competition family, shared by every edition; always nu |
| game_statistic_status_code | character | Box-score validation status: VALID. |
| game_play_by_play_status_code | character | Play-by-play availability flag: VALID or EMPTY. UNI |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, overtime remark); null wh |
| window_code | character | Qualifier window code; null otherwise. |
| window_name | character | Qualifier window name; null otherwise. |
| team_a_from | character | How team A qualified for this slot ('Winner of Game 8 |
| team_b_from | character | How team B qualified for this slot ('Winner of Game 8 |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`, `fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_stats()`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

---

`fiba_gdap_gateway_competition_ids`

# FIBA GDAP: gateway competition ids

---

## Description

Ids of the competitions that run in a date window.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getcustomgateway`

## Usage

```
fiba_gdap_gateway_competition_ids(start_date, end_date, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `start_date` | (character, required) Required. Window start as YYYY-MM-DD. |
| `end_date` | (character, required) Required. Window end as YYYY-MM-DD. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_gateway_competition_ids(start_date = "2025-01-14", end_date = "2025-01-16")
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64, GDAP key space) of a competition that runs in the window. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_federation_ranking()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_stats()`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

`fiba_gdap_gateway_federation_ranking`

**FIBA GDAP: gateway federation ranking**

## Description

FIBA world ranking from the server-side gateway (same items as fiba_gdap_federation_ranking).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getcustomgateway`

## Usage

```
fiba_gdap_gateway_federation_ranking(
  gender = "women",
  zone_id = 1L,
  as_of_date = NULL,
  return_parsed = TRUE
)
```

## Arguments

| | |
|---|---|
| `gender` | (character, default `"women"` in wehoop; sdv-py and hoopR default `"men"`) federationRanking: men or women (= gdapCategory) |
| `zone_id` | (integer, default `1L`) federationRanking: ranking zone id (= zoneRankingId) |
| `as_of_date` | (character, optional) Ranking as of this date, ISO 8601: the latest ranking published on or before it (2026-03-10 answers the 2026-03-03 ranking). Omitted = the latest ranking |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_gateway_federation_ranking(gender = "women", zone_id = 1, as_of_date = "2025-09-15")
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| as_of_date | character | Date of the ranking returned: the latest ranking published on or before the requeste |
| zone_ranking_code | character | Ranking zone code (WORLD, AFRICA...). |
| zone_ranking_official_name | character | Ranking zone official name. |
| world_rank | integer | Position in the FIBA world ranking (1 = best). |
| country_name | character | Country name in English. |
| zone_rank | integer | Rank within the zone. |
| ioc_code | character | IOC country code. |
| fiba_code | character | FIBA three-letter country code (GER, USA; historic URS, YUG). |
| current_points | double | Ranking points (one decimal). |
| world_rank_variation | integer | Change in world rank since the previous release (positive = up). |
| comments | character | Ranking note; null in every sample. |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for |
| organisation_status_code | character | Organisation status: ACT active. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_stats()`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

`fiba_gdap_medal_count` **FIBA GDAP: medal count**

## Description

All-time medal table of a competition family.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapmedalcount`

## Usage

```
fiba_gdap_medal_count(family_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `family_id` | (integer, required) Required. Competition family id (competitionMarketingName.competitionMarketingNameId): 201 = World Cup, 320 = men's Olympics, 306 = Women's World Cup, 112 = BCL tier 1, 211 = EuroLeague Women. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_medal_count(family_id = 201)
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| team_name | character | Team name (country or club). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nationa |
| fiba_code | character | FIBA three-letter country code (GER, USA; historic URS, YUG). |
| team_zone_code | character | Zone code of the team (E Europe, NAM Americas north...). |
| rank | integer | Rank (1 = first). |
| total_first_rank | integer | Gold medals (first places). |
| total_second_rank | integer | Number of silver medals the federation has won in the competition family. |
| total_third_rank | integer | Number of bronze medals the federation has won in the competition family. |
| total_participations | integer | Editions played. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

fiba_gdap_national_teams

# FIBA GDAP: national teams

---

## Description

Every national federation FIBA lists.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapnationalteams

## Usage

```
fiba_gdap_national_teams(return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_national_teams()
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| national_federation_id | integer | Federation organisation id (organisationId key space: 64 = Germany). |
| display_name | character | English display name of the federation (the country), as the FIBA site shows it. |
| official_name | character | Federation's official name. |
| country_id | integer | FIBA country id (country table key space; not an organisationId). |
| country_name | character | Country name in English. |
| country_fiba_code | character | FIBA code of the federation's country. |
| zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone table key space). |
| zone_name | character | FIBA zone (Europe, Africa...). |
| organisation_status_code | character | Organisation status: ACT active. |
| membership_status_code | character | FIBA membership status: ACT active. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`

fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(), fiba_gdap_medal_count(), fiba_gdap_nominations(), fiba_gdap_player_game_stats(), fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_nominations` **FIBA GDAP: nominations**

---

## Description

Officials nominated for a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionnominationsbyid`

## Usage

```
fiba_gdap_nominations(comp, level = "competition", return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `level` | (character, default `"competition"`) competition = officials nominated for the whole competition (the only value the site sends) |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_nominations(comp = 208182)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_level | character | Level the nomination is attached to: Competition. |
| competition_level_id | integer | Id of the object the nomination is attached to: the competitionId w |
| competition_role | character | Role label at competition level; null in every sample. |
| role | character | Role: Referee, FIBA Table Officials, FIBA Certified Statisticians, |
| role_code | character | Role code (REF, FIBATDEL...). |
| game_assignment_role | character | Role on a game assignment; null at competition level. |
| game_assignment_role_code | character | Code of gameAssignmentRole; null at competition level. |
| game_assignment_role_order | character | Order of gameAssignmentRole; null at competition level. |
| person_id | integer | FIBA person id, global across competitions and roles (player, coac |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnc |
| nationalities | character | Nationalities (FIBA codes on player info; country objects on roste |
| license_nationality | character | FIBA code of the federation that licenses the official. |
| affiliated_national_federation_country_code | character | FIBA code of the federation the official is affiliated to. |
| affiliated_national_federation_country_name | character | Country of the federation the official is affiliated to. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`
`fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_player_game_stats()`, `fiba_gdap_player_info()`, `fiba_gdap_player_stats()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`, `fiba_gdap_rounds()`, `fiba_gdap_schedule()`, `fiba_gdap_team_games()`, `fiba_gdap_team_player_stats()`, `fiba_gdap_team_staff()`, `fiba_gdap_team_stats()`, `fiba_gdap_teams()`

`fiba_gdap_player_game_stats`

## FIBA GDAP: player game stats

### Description

A player's game log (full box lines) at a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapplayergamestatisticsbyplayerid`

### Usage

```
fiba_gdap_player_game_stats(player_id, comp, return_parsed = TRUE)
```

### Arguments

| | |
|---|---|
| `player_id` | (integer, required) Required. FIBA person id (global): personId of a roster row, pId in play-by-play, P_<id> in the game-page box |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

### Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| player_id | integer | FIBA person id (Int64), global; the player_id argument. |
| competition_id | integer | FIBA competition id (Int64) the game log belongs to; the comp argument. |
| game_id | integer | FIBA game id, global across competitions (GDAP key space): gdapGameId, the < |
| game_day | integer | Game day number within the round. |
| round_id | integer | Round id (GDAP key space, unique across competitions); the first segment of gam |
| round_code | character | Round code (GP group phase, QF, SF, F final, RS regular season, CL17-32 classifi |
| round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 202 |
| team_name | character | Player's team name. |
| team_code | character | Player's team code. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation fo |
| versus_team_id | integer | Opponent's competition team id (see teamId). |
| versus_team_name | character | Opponent's name. |
| versus_team_code | character | Opponent's code. |
| versus_team_organisation_id | integer | Opponent's organisation id (see organisationId). |
| play_duration_in_seconds | integer | Playing time in seconds in the game. |
| points | integer | Points in the game. |
| field_goals_made | integer | Field goals made in the game. |
| field_goals_attempted | integer | Field goals attempted in the game. |
| field_goals_percentage | double | Field-goal percentage, 0-100 (in the game; full float precision in averages). |
| two_points_made | integer | Two-point field goals made in the game. |
| two_points_attempted | integer | Two-point field goals attempted in the game. |
| two_points_percentage | double | Two-point field-goal percentage, 0-100 (in the game; full float precision in averag |
| three_points_made | integer | Three-point field goals made in the game. |
| three_points_attempted | integer | Three-point field goals attempted in the game. |
| three_points_percentage | double | Three-point field-goal percentage, 0-100 (in the game; full float precision in avera |
| free_throws_made | integer | Free throws made in the game. |
| free_throws_attempted | integer | Free throws attempted in the game. |
| free_throws_percentage | double | Free-throw percentage, 0-100 (in the game; full float precision in averages). |
| offensive_rebounds | integer | Offensive rebounds in the game. |
| defensive_rebounds | integer | Defensive rebounds in the game. |
| rebounds | integer | Rebounds in the game. |
| assists | integer | Assists in the game. |
| personal_fouls | integer | Personal fouls in the game. |
| turnovers | integer | Turnovers in the game. |
| steals | integer | Steals in the game. |
| blocked_shots | integer | Blocked shots in the game. |
| plus_minus | integer | Plus/minus (point differential while on court) in the game. |
| efficiency | integer | Efficiency (FIBA EFF) in the game. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`,

fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank
fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(),
fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(
fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_info(),
fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(),
fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(),
fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(),
fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_player_game_stats(player_id = 211501, comp = 208182))
```

---

# fiba_gdap_player_info FIBA GDAP: player info

---

## Description

A player's bio and per-game averages with ranks at a competition.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapplayerinfobycompetitionid

## Usage

```
fiba_gdap_player_info(player_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| player_id | (integer, required) Required. FIBA person id (global): personId of a roster row, pId in play-by-play, P_<id> in the game-page box |
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_player_info(player_id = 211501, comp = 208182)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| player_id | integer | FIBA person id, global across competitions and roles (player, c |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 |
| competition_team_id | integer | Competition team id: one per team per competition (Germany = |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the nat |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| height_cm | integer | Height in centimetres. |
| height_ft_in | character | Height, feet and inches string. |
| club_name | character | Club at the time of the competition. |
| club_code | character | Club abbreviation; always null in the samples, values unverified |
| is_player_free_agent | logical | True when the player had no club. |
| points_per_game | double | The player's average points per game played. |
| rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| rebounds_per_game | double | The player's average rebounds per game played. |
| rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| assists_per_game | double | The player's average assists per game played. |
| rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| points_per_game_per_competition_category | double | Average points per game across the competition category, the c |
| rebounds_per_game_per_competition_category | double | Average rebounds per game across the competition category, th |
| assists_per_game_per_competition_category | double | Average assists per game across the competition category, the c |
| birth_date | character | Date of birth, YYYY-MM-DD. |
| nationalities | character | Nationalities (FIBA codes on player info; country objects on ro |
| social_media_website | character | Website URL; always null in the samples, values unverified. |
| social_media_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| social_media_facebook | character | Facebook handle or URL; always null in the samples, values un |
| social_media_instagram | character | Instagram handle; always null in the samples, values unverified. |
| social_media_youtube | character | YouTube handle; always null in the samples, values unverified. |
| social_networks | character | Social-network handles. Stored as a JSON string. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_player_stats`

# FIBA GDAP: player stats

---

## Description

A player's per-game averages, totals and highs at a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapplayercompetitionstatisticsbyplayerid`

## Usage

```
fiba_gdap_player_stats(player_id, comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `player_id` | (integer, required) Required. FIBA person id (global): personId of a roster row, pId in play-by-play, P_<id> in the game-page box |
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| player_id | integer | FIBA person id, global across competitions and roles (player, coach, official): r |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = 2( |
| points_per_game | double | The player's average points per game played. |
| rebounds_per_game | double | The player's average rebounds per game played. |
| rebounds_offensive_per_game | double | The player's average offensive rebounds per game played. |
| rebounds_defensive_per_game | double | The player's average defensive rebounds per game played. |
| assists_per_game | double | The player's average assists per game played. |
| blocks_per_game | double | The player's average blocked shots per game played. |
| turnovers_per_game | double | The player's average turnovers per game played. |
| fouls_per_game | double | The player's average personal fouls per game played. |
| plus_minus_per_game | double | The player's average plus/minus (point differential while on court) per game pl |
| efficiency_per_game | double | The player's average efficiency (FIBA EFF) per game played. |
| play_time_in_seconds_per_game | double | The player's average playing time in seconds per game played. |
| field_goals_made_per_game | double | The player's average field goals made per game played. |
| field_goals_attempted_per_game | double | The player's average field goals attempted per game played. |
| two_points_made_per_game | double | The player's average two-point field goals made per game played. |
| two_points_attempted_per_game | double | The player's average two-point field goals attempted per game played. |
| three_points_made_per_game | double | The player's average three-point field goals made per game played. |
| three_points_attempted_per_game | double | The player's average three-point field goals attempted per game played. |
| free_throws_made_per_game | double | The player's average free throws made per game played. |
| free_throws_attempted_per_game | double | The player's average free throws attempted per game played. |
| points_from_turnovers_per_game | double | The player's average points from turnovers per game played. |
| fast_break_points_per_game | double | The player's average fast-break points per game played. |
| second_chance_points_per_game | double | The player's average second-chance points per game played. |
| points_in_the_paint_per_game | double | The player's average points in the paint per game played. |
| steals_per_game | double | The player's average steals per game played. |
| total_games_played | integer | The player's total games played over the games played. |

| | | |
|---|---|---|
| total_games_won | integer | The player's total games won over the games played. |
| total_games_lost | integer | The player's total games lost over the games played. |
| total_rebounds | integer | The player's total rebounds over the games played. |
| total_rebounds_offensive | integer | The player's total offensive rebounds over the games played. |
| total_rebounds_defensive | integer | The player's total defensive rebounds over the games played. |
| total_blocks | integer | The player's total blocked shots over the games played. |
| total_turnovers | integer | The player's total turnovers over the games played. |
| total_fouls | integer | The player's total personal fouls over the games played. |
| total_assists | integer | The player's total assists over the games played. |
| total_plus_minus | integer | The player's total plus/minus (point differential while on court) over the games |
| total_efficiency | integer | The player's total efficiency (FIBA EFF) over the games played. |
| total_play_time_in_seconds | integer | The player's total playing time in seconds over the games played. |
| total_points | integer | The player's total points over the games played. |
| total_field_goals_made | integer | The player's total field goals made over the games played. |
| total_field_goals_attempted | integer | The player's total field goals attempted over the games played. |
| field_goals_percentage | double | Field-goal percentage, 0-100 (over the competition; full float precision in avera |
| total_two_points_made | integer | The player's total two-point field goals made over the games played. |
| total_two_points_attempted | integer | The player's total two-point field goals attempted over the games played. |
| two_points_percentage | double | Two-point field-goal percentage, 0-100 (over the competition; full float precisio |
| total_three_points_made | integer | The player's total three-point field goals made over the games played. |
| total_three_points_attempted | integer | The player's total three-point field goals attempted over the games played. |
| three_points_percentage | double | Three-point field-goal percentage, 0-100 (over the competition; full float precis |
| total_free_throws_made | integer | The player's total free throws made over the games played. |
| total_free_throws_attempted | integer | The player's total free throws attempted over the games played. |
| free_throws_percentage | double | Free-throw percentage, 0-100 (over the competition; full float precision in aver |
| total_double_doubles | integer | The player's total double-doubles over the games played. |
| total_triple_doubles | integer | The player's total triple-doubles over the games played. |
| total_quadruple_doubles | integer | The player's total quadruple-doubles over the games played. |
| total_quintuple_doubles | integer | The player's total quintuple-doubles over the games played. |
| total_points_from_turnovers | integer | The player's total points from turnovers over the games played. |
| total_fast_break_points | integer | The player's total fast-break points over the games played. |
| total_second_chance_points | integer | The player's total second-chance points over the games played. |
| total_points_in_the_paint | integer | The player's total points in the paint over the games played. |
| total_steals | integer | The player's total steals over the games played. |
| rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`
`fiba_gdap_medal_count()`, `fiba_gdap_national_teams()`, `fiba_gdap_nominations()`, `fiba_gdap_player_game_sta`
`fiba_gdap_player_info()`, `fiba_gdap_ranking_zones()`, `fiba_gdap_roster()`, `fiba_gdap_roster_coaches()`, `fiba_gdap_roster_latest()`, `fiba_gdap_round_gamedays()`, `fiba_gdap_round_standings()`,

fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_player_stats(player_id = 211501, comp = 208182))
```

---

`fiba_gdap_ranking_zones`

**FIBA GDAP: ranking zones**

---

## Description

FIBA world-ranking zones.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdaprankingzones`

## Usage

```
fiba_gdap_ranking_zones(return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_ranking_zones()
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| zone_ranking_id | integer | Ranking zone id (getgdaprankingzones key space: 1 = World ... 5 = Europe). |
| code | character | Ranking zone code (W, AF, AM, AS, EU). |
| official_name | character | Official name of the ranking zone (World, FIBA Europe, FIBA Americas...). |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

# fiba_gdap_roster FIBA GDAP: roster

---

## Description

A team's roster at a competition.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamrosterbyteamid

## Usage

```
fiba_gdap_roster(team_id, discipline_format = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| team_id | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| discipline_format | (character, optional) Optional; the site passes its discipline format. 5x5 answers the same as omitting it (verified on awards and rosters) |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team code (FIBA country code or club abbreviation). |
| team_name | character | Team name (the country for a national team, the club for a club team). |
| organisation_id | integer | Stable federation or club id (Int64), the same in every competition and season |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| person_id | integer | FIBA person id, global across competitions and roles (player, coach, official): |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnominationsb |
| date_of_birth | character | Date of birth, YYYY-MM-DD. |
| uniform_number | character | Jersey number (a string). |
| uniform_name | character | Name printed on the jersey. |
| position | character | Playing position (roster: Point Guard...; game page: PG, SG, SF, PF, C). |
| height_in_cm | integer | Player height in centimetres. |
| height_in_feet_inches | character | Height, feet and inches string (6'7"). |
| nationality | character | FIBA code of the main nationality. |
| is_on_final_roster | logical | True when on the final (12-player) roster. |
| games_played | integer | Games played in the competition. |
| club_id | integer | Organisation id of the player's club (organisationId key space). |

| | | |
|---|---|---|
| club_name | character | Club at the time of the competition. |
| country_fiba_code | character | FIBA code of the country the person represents. |
| club_country_fiba_code | character | FIBA code of the club's country. |
| is_captain | logical | True for the team captain. |
| is_player_free_agent | logical | True when the player had no club. |
| final_roster_member_status_code | character | Roster entry status: VALID, PARTIAL (incomplete registration). |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster_coaches( fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_roster(team_id = 279569))
```

---

`fiba_gdap_roster_coaches`

# FIBA GDAP: roster coaches

---

## Description

A team's coaches on its roster at a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamrosterbyteamid`

## Usage

```
fiba_gdap_roster_coaches(
  team_id,
  discipline_format = NULL,
  return_parsed = TRUE
)
```

## Arguments

| | |
|---|---|
| team_id | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| discipline_format | (character, optional) Optional; the site passes its discipline format. 5x5 answers the same as omitting it (verified on awards and rosters) |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_roster_coaches(team_id = 279569)
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team code (FIBA country code or club abbreviation). |
| team_name | character | Team name (the country for a national team, the club for a club team). |
| organisation_id | integer | Stable federation or club id (Int64), the same in every competition and seasor |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| person_id | integer | FIBA person id, global across competitions and roles (player, coach, official) |
| first_name | character | Given name(s) as registered with FIBA. |

| | | |
|---|---|---|
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnominationsl |
| function | character | Role: Head Coach, Coach, Assistant Coach. |
| is_head_coach | integer | 1 for the head coach, 0 otherwise (an integer flag, not a boolean). |
| is_assistant_coach | integer | 1 for an assistant coach, else null (integer flag). |
| is_accompanying_coach | logical | Accompanying-coach flag; always null in the samples, values unverified. |
| coach_display_order | integer | Display order; null when unset. |
| is_on_final_roster | logical | True when on the final (12-player) roster. |
| nationalities | character | Nationalities (FIBA codes on player info; country objects on rosters and offic |
| coach_license_organisation_name | character | Federation that issued the coaching licence. |
| coach_license_country_code | character | FIBA code of the licensing federation's country. |
| final_roster_member_status_code | character | Roster entry status: VALID, PARTIAL (incomplete registration). |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_roster_latest`

# FIBA GDAP: roster latest

---

## Description

A team's latest roster (same shape as the roster).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamlatestrosterbyteamid`

## Usage

```
fiba_gdap_roster_latest(team_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `team_id` | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_roster_latest(team_id = 282183)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team code (FIBA country code or club abbreviation). |
| team_name | character | Team name (the country for a national team, the club for a club team). |
| organisation_id | integer | Stable federation or club id (Int64), the same in every competition and season |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| person_id | integer | FIBA person id, global across competitions and roles (player, coach, official): |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnominationsb |
| date_of_birth | character | Date of birth, YYYY-MM-DD. |
| uniform_number | character | Jersey number (a string). |
| uniform_name | character | Name printed on the jersey. |
| position | character | Playing position (roster: Point Guard...; game page: PG, SG, SF, PF, C). |
| height_in_cm | integer | Player height in centimetres. |
| height_in_feet_inches | character | Height, feet and inches string (6'7"). |
| nationality | character | FIBA code of the main nationality. |
| is_on_final_roster | logical | True when on the final (12-player) roster. |

| | | |
|---|---|---|
| games_played | integer | Games played in the competition. |
| club_id | integer | Organisation id of the player's club (organisationId key space). |
| club_name | character | Club at the time of the competition. |
| country_fiba_code | character | FIBA code of the country the person represents. |
| club_country_fiba_code | character | FIBA code of the club's country. |
| is_captain | logical | True for the team captain. |
| is_player_free_agent | logical | True when the player had no club. |
| final_roster_member_status_code | character | Roster entry status: VALID, PARTIAL (incomplete registration). |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_round_gamedays`

# FIBA GDAP: round gamedays

---

## Description

Rounds of a competition with their game days.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionroundgamedaysbyid`

## Usage

```
fiba_gdap_round_gamedays(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_round_gamedays(comp = 208182)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the rounds belong to; the comp argument. |
| round_game_system | character | Pairing system label (Single, Home and Away, Best of 3). |
| round_game_system_code | character | Pairing system code: S single game, HA home and away, BOF3 best of three |
| round_parent_round_id | integer | roundId of the round this one belongs to; null for a top-level round. |
| round_round_code | character | Round code (GP group phase, QF, SF, F final, RS regular season, CL17-32 c |
| round_round_id | integer | Round id (GDAP key space, unique across competitions); the first segment c |
| round_round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| round_round_number | integer | Round number within the competition (1-based). |
| round_round_type | character | Round type label: Group Round, Non Group Round, Super Round (getgdapg |
| round_round_type_code | character | Round type code: G group, NG non-group (knockout), S super round. |
| round_round_status_code | character | Round status: COMPL completed, PROGR in progress, CLOS closed. |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59:59.9999999 on the last day). |
| round_has_teams_assigned | logical | True once the round's teams are known. |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_standing_header_code | character | Code of the standings table the round feeds. |
| round_standing_header_name | character | Name of the standings table the round feeds. |

| | | |
|---|---|---|
| round_qualified_team_information | character | Qualification note of the round; null in every sample. |
| round_round_order | integer | Display order of the round. |
| gamedays | character | Game days of the round as a JSON string; empty (null) for the competitions |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

`fiba_gdap_round_standings`

# FIBA GDAP: round standings

---

## Description

Group standings of every round of a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdaproundsbycompetitionid`

## Usage

```
fiba_gdap_round_standings(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| round_id | integer | Round id (Int64) the group belongs to. |
| round_name | character | Round name (Preliminary Round, Final Round...). |
| round_code | character | Round code (a short code such as G or F). |
| group_name | character | Group name within the round (A, B...). |
| group_status_code | character | Group status code (COMPL when the group is complete). |
| rank | integer | Rank in the group. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |
| team_code | character | Team code (FIBA country code or club abbreviation). |
| team_official_name | character | Official team name (the country for a national team, the club for a club team). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| games_played | integer | Games played in the group. |
| games_lost | integer | Losses in the group. |
| games_won | integer | Wins in the group. |
| points | integer | Standing points (2 per win, 1 per loss). |
| points_against | integer | Points conceded. |
| points_for | integer | Points the team scored in the group's games. |
| starting_position | integer | Seed / draw position in the group. |
| final_position | integer | Final position in the group. |
| plus_minus | integer | Point differential (pointsFor - pointsAgainst). |
| penalty_points | integer | Standing points deducted. |
| draw_points | integer | Points for a draw (0); null when not applicable. |
| group_game_results | character | This team's games in the group. Stored as a JSON string. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank
fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(
fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta
fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_round_standings(comp = 208182))
```

---

`fiba_gdap_rounds` **FIBA GDAP: rounds**

---

## Description

Rounds of a competition (group stage, knockout rounds).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdaproundsbycompetitionid`

## Usage

```
fiba_gdap_rounds(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_system | character | Pairing system label (Single, Home and Away, Best of 3). |
| game_system_code | character | Pairing system code: S single game, HA home and away, BOF3 best of three. |
| parent_round_id | integer | roundId of the round this one belongs to; null for a top-level round. |
| round_code | character | Round code (GP group phase, QF, SF, F final, RS regular season, CL17-32 classific |
| round_id | integer | Round id (GDAP key space, unique across competitions); the first segment of game |
| round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| round_number | integer | Round number within the competition (1-based). |
| round_type | character | Round type label: Group Round, Non Group Round, Super Round (getgdapgamesb |
| round_type_code | character | Round type code: G group, NG non-group (knockout), S super round. |
| round_status_code | character | Round status: COMPL completed, PROGR in progress, CLOS closed. |
| start_date | character | Round start, local date-time. |
| end_date | character | Round end, local date-time (23:59:59.9999999 on the last day). |
| has_teams_assigned | logical | True once the round's teams are known. |
| round_kind | character | Standard Round or Qualifiers. |
| standing_header_code | character | Code of the standings table the round feeds. |
| standing_header_name | character | Name of the standings table the round feeds. |
| qualified_team_information | character | Qualification note of the round; null in every sample. |
| round_order | integer | Display order of the round. |

## See Also

Other FIBA Functions: `fiba_gamepage_box_players()`, `fiba_gamepage_box_teams()`, `fiba_gamepage_lineup()`, `fiba_gamepage_pbp()`, `fiba_gamepage_period_scores()`, `fiba_gamepage_shots()`, `fiba_gdap_awards()`, `fiba_gdap_competition()`, `fiba_gdap_editions()`, `fiba_gdap_federation_games()`, `fiba_gdap_federation_rank`
`fiba_gdap_final_standings()`, `fiba_gdap_game()`, `fiba_gdap_game_officials()`, `fiba_gdap_game_periods()`, `fiba_gdap_games_between()`, `fiba_gdap_gateway_competition_ids()`, `fiba_gdap_gateway_federation_ranking()`

fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta
fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(),
fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(),
fiba_gdap_round_standings(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(
fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_rounds(comp = 208182))
```

---

`fiba_gdap_schedule` **FIBA GDAP: schedule**

---

## Description

Schedule and results of one competition (edition): one row per game.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapgamesbycompetitionid`

## Usage

```
fiba_gdap_schedule(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId: |
| game_number | character | Game number within the round or |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tear |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cour |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tear |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cour |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is known |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |

| | | |
|---|---|---|
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gan |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |

| | | |
|---|---|---|
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slo |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_schedule(comp = 208182))
```

---

# fiba_gdap_team_games FIBA GDAP: team games

---

## Description

A team's games with each team's leaders.

Endpoint: GET https://digital-api.fiba.basketball/hapi/getgdapgameswithleaderdetailsbyteamid

## Usage

```
fiba_gdap_team_games(team_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| team_id | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across competitions (GDAP key space) |
| round_name | character | Round label (Group Phase, Quarter-Finals, Final...). |
| game_number | character | Game number within the round or pairing (a string: '1', 'A'). |
| status | character | Game status label (Validated). |
| status_code | character | Game status code: VALID (validated result), CLOS (closed, |
| game_day | integer | Game day number within the round. |
| team_a_from | character | How team A qualified for this slot; null in group play. |
| team_b_from | character | How team B qualified for this slot; null in group play. |
| team_a_team_id | integer | Competition team id: one per team per competition (Germany |
| team_a_organisation_id | integer | Organisation id, stable across competitions and seasons: the r |
| team_a_code | character | Team code (national: FIBA country code; club: a club abbrev |
| team_a_official_name | character | Official name of team A: the country for a national team, the |
| team_a_short_name | character | Short name of team A: the country code for a national team, t |
| team_b_team_id | integer | Competition team id: one per team per competition (Germany |
| team_b_organisation_id | integer | Organisation id, stable across competitions and seasons: the r |
| team_b_code | character | Team code (national: FIBA country code; club: a club abbrev |
| team_b_official_name | character | Official name of team B: the country for a national team, the |
| team_b_short_name | character | Short name of team B: the country code for a national team, t |
| team_a_score | integer | Team A points in the period. |

| | | |
|---|---|---|
| team_b_score | integer | Team B points in the period. |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FIBA code in getgdapgamesw |
| host_country_code | character | FIBA code of the venue's country; always null in the samples |
| venue_id | integer | Venue id (GDAP key space; 0 = unknown). |
| venue_name | character | Name of the arena the game is played in. |
| game_date_time | character | Tip-off in venue-local wall-clock time, ISO 8601 without offs |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 without the Z. |
| has_time_game_date_time | logical | False when only the date is known (the time part is then 00:0 |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in hours (Float64: a half-hour zo |
| buy_tickets_url | character | Ticket link; always '' in the samples. |
| watch_url | character | Stream link; always '' in the samples. |
| is_postponed | logical | True when the game was postponed. |
| is_played_behind_closed_doors | logical | True when played without spectators. |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as recorded by FIBA. |
| is_live | logical | True while the game is being played. |
| leaders_team_a_points_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_a_points_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_a_points_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_a_points_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_a_points_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_a_points_leader_rank | integer | Rank (1 = first). |
| leaders_team_a_rebounds_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_a_rebounds_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_a_rebounds_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_a_rebounds_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_a_rebounds_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_a_rebounds_leader_rank | integer | Rank (1 = first). |
| leaders_team_a_assists_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_a_assists_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_a_assists_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_a_assists_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_a_assists_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_a_assists_leader_rank | integer | Rank (1 = first). |
| leaders_team_b_points_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_b_points_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_b_points_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_b_points_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_b_points_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_b_points_leader_rank | integer | Rank (1 = first). |
| leaders_team_b_rebounds_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_b_rebounds_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_b_rebounds_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_b_rebounds_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_b_rebounds_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_b_rebounds_leader_rank | integer | Rank (1 = first). |

| | | |
|---|---|---|
| leaders_team_b_assists_leader_player_id | integer | FIBA person id, global across competitions and roles (player, |
| leaders_team_b_assists_leader_first_name | character | Given name(s) as registered with FIBA. |
| leaders_team_b_assists_leader_last_name | character | Last name (officials: upper case). |
| leaders_team_b_assists_leader_statistics_value | integer | Leader's value for the statistic. |
| leaders_team_b_assists_leader_nationality | character | FIBA code of the main nationality. |
| leaders_team_b_assists_leader_rank | integer | Rank (1 = first). |
| periods | character | Score by period. Stored as a JSON string. |
| game_statistic_status_code | character | Box-score validation status: VALID. |
| game_play_by_play_status_code | character | Play-by-play availability flag: VALID or EMPTY. UNRELIA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, overtime remark); null when no |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_team_games(team_id = 279569))
```

---

`fiba_gdap_team_player_stats`

# FIBA GDAP: team player stats

---

## Description

Per-game averages of every player of a team at a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamstatisticsbyteamid`

## Usage

```
fiba_gdap_team_player_stats(team_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| team_id | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_team_player_stats(team_id = 279569)
```

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the averages belong to. |
| player_id | integer | FIBA person id, global across competitions and roles (player, coach, official) |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| uniform_number | character | Jersey number (a string). |
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| points_per_game | double | The player's average points per game played. |
| rebounds_per_game | double | The player's average rebounds per game played. |
| rebounds_offensive_per_game | double | The player's average offensive rebounds per game played. |
| rebounds_defensive_per_game | double | The player's average defensive rebounds per game played. |

| | | |
|---|---|---|
| assists_per_game | double | The player's average assists per game played. |
| blocks_per_game | double | The player's average blocked shots per game played. |
| turnovers_per_game | double | The player's average turnovers per game played. |
| fouls_per_game | double | The player's average personal fouls per game played. |
| plus_minus_per_game | double | The player's average plus/minus (point differential while on court) per game |
| efficiency_per_game | double | The player's average efficiency (FIBA EFF) per game played. |
| play_time_in_seconds_per_game | double | The player's average playing time in seconds per game played. |
| field_goals_made_per_game | double | The player's average field goals made per game played. |
| field_goals_attempted_per_game | double | The player's average field goals attempted per game played. |
| two_points_made_per_game | double | The player's average two-point field goals made per game played. |
| two_points_attempted_per_game | double | The player's average two-point field goals attempted per game played. |
| three_points_made_per_game | double | The player's average three-point field goals made per game played. |
| three_points_attempted_per_game | double | The player's average three-point field goals attempted per game played. |
| free_throws_made_per_game | double | The player's average free throws made per game played. |
| free_throws_attempted_per_game | double | The player's average free throws attempted per game played. |
| points_from_turnovers_per_game | double | The player's average points from turnovers per game played. |
| fast_break_points_per_game | double | The player's average fast-break points per game played. |
| second_chance_points_per_game | double | The player's average second-chance points per game played. |
| points_in_the_paint_per_game | double | The player's average points in the paint per game played. |
| steals_per_game | double | The player's average steals per game played. |
| total_games_played | integer | The player's total games played over the games played. |
| total_games_won | integer | The player's total games won over the games played. |
| total_games_lost | integer | The player's total games lost over the games played. |
| total_rebounds | integer | The player's total rebounds over the games played. |
| total_rebounds_offensive | integer | The player's total offensive rebounds over the games played. |
| total_rebounds_defensive | integer | The player's total defensive rebounds over the games played. |
| total_blocks | integer | The player's total blocked shots over the games played. |
| total_turnovers | integer | The player's total turnovers over the games played. |
| total_fouls | integer | The player's total personal fouls over the games played. |
| total_assists | integer | The player's total assists over the games played. |
| total_plus_minus | integer | The player's total plus/minus (point differential while on court) over the gam |
| total_efficiency | integer | The player's total efficiency (FIBA EFF) over the games played. |
| total_play_time_in_seconds | integer | The player's total playing time in seconds over the games played. |
| total_points | integer | The player's total points over the games played. |
| total_field_goals_made | integer | The player's total field goals made over the games played. |
| total_field_goals_attempted | integer | The player's total field goals attempted over the games played. |
| field_goals_percentage | double | Field-goal percentage, 0-100 (over the competition; full float precision in ave |
| total_two_points_made | integer | The player's total two-point field goals made over the games played. |
| total_two_points_attempted | integer | The player's total two-point field goals attempted over the games played. |
| two_points_percentage | double | Two-point field-goal percentage, 0-100 (over the competition; full float preci |
| total_three_points_made | integer | The player's total three-point field goals made over the games played. |
| total_three_points_attempted | integer | The player's total three-point field goals attempted over the games played. |
| three_points_percentage | double | Three-point field-goal percentage, 0-100 (over the competition; full float pre |
| total_free_throws_made | integer | The player's total free throws made over the games played. |
| total_free_throws_attempted | integer | The player's total free throws attempted over the games played. |
| free_throws_percentage | double | Free-throw percentage, 0-100 (over the competition; full float precision in av |
| total_double_doubles | integer | The player's total double-doubles over the games played. |
| total_triple_doubles | integer | The player's total triple-doubles over the games played. |

| | | |
|---|---|---|
| total_quadruple_doubles | integer | The player's total quadruple-doubles over the games played. |
| total_quintuple_doubles | integer | The player's total quintuple-doubles over the games played. |
| total_points_from_turnovers | integer | The player's total points from turnovers over the games played. |
| total_fast_break_points | integer | The player's total fast-break points over the games played. |
| total_second_chance_points | integer | The player's total second-chance points over the games played. |
| total_points_in_the_paint | integer | The player's total points in the paint over the games played. |
| total_steals | integer | The player's total steals over the games played. |
| rank | integer | Rank (1 = first). |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_staff(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

# fiba_gdap_team_staff **FIBA GDAP: team staff**

---

## Description

Coaches of every team of a competition.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapteamstaffbycompetitionid`

## Usage

```
fiba_gdap_team_staff(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| comp | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the `<gameId>` of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_`<id>` and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

```
fiba_gdap_team_staff(comp = 208182)
```

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | FIBA team id (Int64), one per team per competition; the team the coach is re |
| team_name | character | Team name the coach is registered with. |
| team_code | character | Team code (GER, USA...) the coach is registered with. |
| person_id | integer | FIBA person id, global across competitions and roles (player, coach, official) |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name (officials: upper case). |
| gender | character | Sex of the person: M or F (Male/Female on getgdapcompetitionnominationsl |
| function | character | Role: Head Coach, Coach, Assistant Coach. |
| is_head_coach | integer | 1 for the head coach, 0 otherwise (an integer flag, not a boolean). |
| is_assistant_coach | integer | 1 for an assistant coach, else null (integer flag). |
| is_accompanying_coach | logical | Accompanying-coach flag; always null in the samples, values unverified. |
| coach_display_order | integer | Display order; null when unset. |
| is_on_final_roster | logical | True when on the final (12-player) roster. |
| nationalities | character | Nationalities (FIBA codes on player info; country objects on rosters and offic |
| coach_license_organisation_name | character | Federation that issued the coaching licence. |
| coach_license_country_code | character | FIBA code of the licensing federation's country. |
| final_roster_member_status_code | character | Roster entry status; always null in the samples, values unverified. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_stats(), fiba_gdap_teams()

---

# fiba_gdap_team_stats FIBA GDAP: team stats

---

## Description

A team's per-game averages and totals at a competition, with the bench totals.

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamstatisticsbyteamid`

## Usage

```
fiba_gdap_team_stats(team_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| team_id | (integer, required) Required. Competition team id: one per team per competition (teamA.teamId of a game row; Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024). Not the federation organisationId |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit `wehoop_error`): `wehoop_no_data` when FIBA has nothing (HTTP 404, or GDAP answers `{}` / `[]` for an unknown id); `wehoop_bad_request` for HTTP 400 / 422 or a malformed id, season or date; `wehoop_asset_fetch` for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A `wehoop_data` tibble (the raw JSON list when `return_parsed = FALSE`) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at |
| competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup |
| points_per_game | double | The team's average points per game played. |
| rebounds_per_game | double | The bench players' average rebounds per game played. |
| rebounds_offensive_per_game | double | The bench players' average offensive rebounds per game played. |
| rebounds_defensive_per_game | double | The bench players' average defensive rebounds per game played. |
| assists_per_game | double | The team's average assists per game played. |
| blocks_per_game | double | The team's average blocked shots per game played. |
| turnovers_per_game | double | The bench players' average turnovers per game played. |
| fouls_per_game | double | The bench players' average personal fouls per game played. |
| plus_minus_per_game | double | The team's average plus/minus (point differential while on court) per game |
| efficiency_per_game | double | The team's average efficiency (FIBA EFF) per game played. |
| play_time_in_seconds_per_game | double | The team's average playing time in seconds per game played. |
| field_goals_made_per_game | double | The team's average field goals made per game played. |
| field_goals_attempted_per_game | double | The team's average field goals attempted per game played. |
| two_points_made_per_game | double | The team's average two-point field goals made per game played. |
| two_points_attempted_per_game | double | The team's average two-point field goals attempted per game played. |
| three_points_made_per_game | double | The team's average three-point field goals made per game played. |
| three_points_attempted_per_game | double | The team's average three-point field goals attempted per game played. |
| free_throws_made_per_game | double | The team's average free throws made per game played. |
| free_throws_attempted_per_game | double | The team's average free throws attempted per game played. |
| points_from_turnovers_per_game | double | The team's average points from turnovers per game played. |
| fast_break_points_per_game | double | The team's average fast-break points per game played. |
| second_chance_points_per_game | double | The team's average second-chance points per game played. |
| points_in_the_paint_per_game | double | The team's average points in the paint per game played. |
| from_bench_points_per_game | double | The team's average points by bench players per game played. |
| steals_per_game | double | The team's average steals per game played. |
| total_games_played | integer | The bench players' total games played over the games played. |
| total_games_won | integer | The team's total games won over the games played. |
| total_games_lost | integer | The team's total games lost over the games played. |

| | | |
|---|---|---|
| total_rebounds | integer | The bench players' total rebounds over the games played. |
| total_rebounds_offensive | integer | The bench players' total offensive rebounds over the games played. |
| total_rebounds_defensive | integer | The bench players' total defensive rebounds over the games played. |
| total_blocks | integer | The team's total blocked shots over the games played. |
| total_turnovers | integer | The bench players' total turnovers over the games played. |
| total_fouls | integer | The bench players' total personal fouls over the games played. |
| total_assists | integer | The team's total assists over the games played. |
| total_plus_minus | integer | The team's total plus/minus (point differential while on court) over the gam |
| total_efficiency | integer | The team's total efficiency (FIBA EFF) over the games played. |
| total_play_time_in_seconds | integer | The team's total playing time in seconds over the games played. |
| total_points | integer | The team's total points over the games played. |
| total_field_goals_made | integer | The team's total field goals made over the games played. |
| total_field_goals_attempted | integer | The team's total field goals attempted over the games played. |
| field_goals_percentage | double | Field-goal percentage, 0-100 (over the competition; full float precision in a |
| total_two_points_made | integer | The team's total two-point field goals made over the games played. |
| total_two_points_attempted | integer | The team's total two-point field goals attempted over the games played. |
| two_points_percentage | double | Two-point field-goal percentage, 0-100 (over the competition; full float prec |
| total_three_points_made | integer | The team's total three-point field goals made over the games played. |
| total_three_points_attempted | integer | The team's total three-point field goals attempted over the games played. |
| three_points_percentage | double | Three-point field-goal percentage, 0-100 (over the competition; full float pr |
| total_free_throws_made | integer | The team's total free throws made over the games played. |
| total_free_throws_attempted | integer | The team's total free throws attempted over the games played. |
| free_throws_percentage | double | Free-throw percentage, 0-100 (over the competition; full float precision in a |
| total_double_doubles | integer | The team's total double-doubles over the games played. |
| total_triple_doubles | integer | The team's total triple-doubles over the games played. |
| total_quadruple_doubles | integer | The team's total quadruple-doubles over the games played. |
| total_quintuple_doubles | integer | The team's total quintuple-doubles over the games played. |
| total_points_from_turnovers | integer | The team's total points from turnovers over the games played. |
| total_fast_break_points | integer | The team's total fast-break points over the games played. |
| total_second_chance_points | integer | The team's total second-chance points over the games played. |
| total_points_in_the_paint | integer | The team's total points in the paint over the games played. |
| total_from_bench_points | integer | The team's total points by bench players over the games played. |
| total_steals | integer | The team's total steals over the games played. |
| biggest_lead | integer | Team's biggest lead in any game of the competition, points. |
| bench_total_games_played | integer | The bench players' total games played over the games played. |
| bench_rebounds_per_game | double | The bench players' average rebounds per game played. |
| bench_rebounds_offensive_per_game | double | The bench players' average offensive rebounds per game played. |
| bench_rebounds_defensive_per_game | double | The bench players' average defensive rebounds per game played. |
| bench_turnovers_per_game | double | The bench players' average turnovers per game played. |
| bench_fouls_per_game | double | The bench players' average personal fouls per game played. |
| bench_total_rebounds | integer | The bench players' total rebounds over the games played. |
| bench_total_rebounds_offensive | integer | The bench players' total offensive rebounds over the games played. |
| bench_total_rebounds_defensive | integer | The bench players' total defensive rebounds over the games played. |
| bench_total_turnovers | integer | The bench players' total turnovers over the games played. |
| bench_total_fouls | integer | The bench players' total personal fouls over the games played. |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank
fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking(
fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta
fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_teams()

## Examples

```
try(fiba_gdap_team_stats(team_id = 279569))
```

---

`fiba_gdap_teams` **FIBA GDAP: teams**

---

## Description

Teams of a competition with their profile (final ranking, per-game averages).

Endpoint: `GET https://digital-api.fiba.basketball/hapi/getgdapcompetitionteamsbycompetitionid`

## Usage

```
fiba_gdap_teams(comp, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `comp` | (integer, required) Required. FIBA competition id (GDAP key space), one per edition: competition.competitionId of any game row (208182 = World Cup 2023, 208722 = Paris 2024 men, 208850 = BCL 2024-25). Find it with fiba_gdap_editions or fiba_gdap_games_between. |
| `return_parsed` | Parse the payload into a tibble (default `TRUE`); `FALSE` returns the raw JSON list. |

## Details

FIBA's terms restrict this source. The GDAP developer-portal terms limit API use to 'Authorized Users' (federations, organisers, FIBA partners) and say any broader use 'is not authorised'; the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers.

The GDAP subscription key is public (fiba.basketball ships it in every page); the runtime reads it from the live page at run time and never logs it. Officials' contact details and coaches' and officials' dates of birth are dropped from every body.

Id key spaces: competition_id is one per edition (2023 World Cup = 208182) and game_id is global across competitions (the <gameId> of a game-page URL). team_id is per team per competition (Germany is 279569 at the 2023 World Cup and 281545 at Paris 2024), while organisation_id is the stable federation or club (64 = Germany, every year); the game page's T_<id> and the play-by-play oId are organisation ids, not team ids.

Errors are classed conditions (all inherit wehoop_error): wehoop_no_data when FIBA has nothing (HTTP 404, or GDAP answers {} / [] for an unknown id); wehoop_bad_request for HTTP 400 / 422 or a malformed id, season or date; wehoop_asset_fetch for any other failed fetch or an answer that is not about the request (an error envelope, HTTP 401 after a key refresh, HTTP 500, a body for another game, competition, team, player, federation, family or season). Seasons are FIBA's own key, which equals the sdv key: the END year for club competitions, the event year for national-team events, the target tournament year for qualifiers.

## Value

A wehoop_data tibble (the raw JSON list when return_parsed = FALSE) with the following columns:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |

| | | |
|---|---|---|
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

## See Also

Other FIBA Functions: fiba_gamepage_box_players(), fiba_gamepage_box_teams(), fiba_gamepage_lineup(), fiba_gamepage_pbp(), fiba_gamepage_period_scores(), fiba_gamepage_shots(), fiba_gdap_awards(), fiba_gdap_competition(), fiba_gdap_editions(), fiba_gdap_federation_games(), fiba_gdap_federation_rank fiba_gdap_final_standings(), fiba_gdap_game(), fiba_gdap_game_officials(), fiba_gdap_game_periods(), fiba_gdap_games_between(), fiba_gdap_gateway_competition_ids(), fiba_gdap_gateway_federation_ranking( fiba_gdap_medal_count(), fiba_gdap_national_teams(), fiba_gdap_nominations(), fiba_gdap_player_game_sta fiba_gdap_player_info(), fiba_gdap_player_stats(), fiba_gdap_ranking_zones(), fiba_gdap_roster(), fiba_gdap_roster_coaches(), fiba_gdap_roster_latest(), fiba_gdap_round_gamedays(), fiba_gdap_round_standings(), fiba_gdap_rounds(), fiba_gdap_schedule(), fiba_gdap_team_games(), fiba_gdap_team_player_stats(), fiba_gdap_team_staff(), fiba_gdap_team_stats()

## Examples

```
try(fiba_gdap_teams(comp = 208182))
```

---

`fiba_wafrobasket_competition`

# FIBA competition: FIBA Women's AfroBasket

---

## Description

Per-competition aliases of the FIBA wrappers for FIBA Women's AfroBasket (GDAP competition family 302), named as in sdv-py. Season rule: event year (2025 = 2025).

## Usage

```
fiba_wafrobasket_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_wafrobasket_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wafrobasket_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (2025 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_wafrobasket_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_wafrobasket_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See fiba_gdap_awards() for the shared notes.

A comp of another competition family, or one GDAP gives no family, is wehoop_asset_fetch; a season with no edition is wehoop_no_data; a season listing several editions, or a game function called with neither season nor comp, is wehoop_bad_request.

```
fiba_wafrobasket_editions(season = 2025)
fiba_wafrobasket_schedule(season = 2025)
```

## Value

A wehoop_data tibble with the columns of the base wrapper (the raw JSON list when return_parsed = FALSE), one table per function:

fiba_wafrobasket_editions():

The columns of fiba_gdap_editions():

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |

| | | |
|---|---|---|
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_wafrobasket_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId> |
| game_number | character | Game number within the round or |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tear |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cour |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tear |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cour |
| team_b_short_name | character | Short name of team B: the country |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1') |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |

| | | |
|---|---|---|
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is known |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gam |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |

| | | |
|---|---|---|
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_wafrobasket_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |

`fiba_wafrobasket_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |

| | | |
|---|---|---|
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_wafrobasket_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |

| | | |
|---|---|---|
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_wafrobasket_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |

| | | |
|---|---|---|
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_wafrobasket_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |

| | | |
|---|---|---|
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not rec |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_wafrobasket_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: `fiba_ecw_competition`, `fiba_elw_competition`, `fiba_wamericup_competition`, `fiba_wasiacup_competition`, `fiba_weurobasket_competition`, `fiba_woly_competition`, `fiba_wwc_competition`

fiba_wamericup_competition

## FIBA competition: FIBA Women's AmeriCup

### Description

Per-competition aliases of the FIBA wrappers for FIBA Women's AmeriCup (GDAP competition family 303), named as in sdv-py. Season rule: event year (2025 = 2025).

### Usage

```
fiba_wamericup_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_wamericup_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wamericup_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

### Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (2025 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_wamericup_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_wamericup_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

### Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised

by FIBA should check the terms before using the wrappers. See `fiba_gdap_awards()` for the shared notes.

A `comp` of another competition family, or one GDAP gives no family, is `wehoop_asset_fetch`; a season with no edition is `wehoop_no_data`; a season listing several editions, or a game function called with neither `season` nor `comp`, is `wehoop_bad_request`.

```
fiba_wamericup_editions(season = 2025)
fiba_wamericup_schedule(season = 2025)
```

## Value

A `wehoop_data` tibble with the columns of the base wrapper (the raw JSON list when `return_parsed = FALSE`), one table per function:

`fiba_wamericup_editions()`:

The columns of `fiba_gdap_editions()`:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_wamericup_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId: |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is known |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |

| | | |
|---|---|---|
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gan |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_wamericup_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a natio |

`fiba_wamericup_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_wamericup_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_wamericup_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |

| | | |
|---|---|---|
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_wamericup_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not re |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_wamericup_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: fiba_ecw_competition, fiba_elw_competition, fiba_wafrobasket_competition, fiba_wasiacup_competition, fiba_weurobasket_competition, fiba_woly_competition, fiba_wwc_competition

---

`fiba_wasiacup_competition`

**FIBA competition: FIBA Women's Asia Cup (Division A)**

---

## Description

Per-competition aliases of the FIBA wrappers for FIBA Women's Asia Cup (Division A) (GDAP competition family 305), named as in sdv-py. Season rule: event year (2025 = 2025).

## Usage

```
fiba_wasiacup_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_wasiacup_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wasiacup_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wasiacup_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wasiacup_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

```
fiba_wasiacup_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wasiacup_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wasiacup_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (2025 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_wasiacup_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_wasiacup_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See fiba_gdap_awards() for the shared notes.

A comp of another competition family, or one GDAP gives no family, is wehoop_asset_fetch; a season with no edition is wehoop_no_data; a season listing several editions, or a game function called with neither season nor comp, is wehoop_bad_request.

```
fiba_wasiacup_editions(season = 2025)
fiba_wasiacup_schedule(season = 2025)
```

## Value

A wehoop_data tibble with the columns of the base wrapper (the raw JSON list when return_parsed = FALSE), one table per function:

fiba_wasiacup_editions():

The columns of fiba_gdap_editions():

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |

| | | |
|---|---|---|
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_wasiacup_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId: |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |

| | | |
|---|---|---|
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is known |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairin |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quarte |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hom |
| round_game_system_code | character | Pairing system code: S single gam |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |

| | | |
|---|---|---|
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_wasiacup_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |

`fiba_wasiacup_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |

| | | |
|---|---|---|
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_wasiacup_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |

| | | |
|---|---|---|
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_wasiacup_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |

| | | |
|---|---|---|
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_wasiacup_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |

| | | |
|---|---|---|
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not rec |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_wasiacup_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: fiba_ecw_competition, fiba_elw_competition, fiba_wafrobasket_competition fiba_wamericup_competition, fiba_weurobasket_competition, fiba_woly_competition, fiba_wwc_competition

---

fiba_weurobasket_competition

# FIBA competition: FIBA Women's EuroBasket

---

## Description

Per-competition aliases of the FIBA wrappers for FIBA Women's EuroBasket (GDAP competition family 297), named as in sdv-py. Season rule: event year (2025 = 2025).

## Usage

```
fiba_weurobasket_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_weurobasket_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_weurobasket_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (2025 = 2025)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_weurobasket_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_weurobasket_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See `fiba_gdap_awards()` for the shared notes.

A `comp` of another competition family, or one GDAP gives no family, is `wehoop_asset_fetch`; a season with no edition is `wehoop_no_data`; a season listing several editions, or a game function called with neither `season` nor `comp`, is `wehoop_bad_request`.

```
fiba_weurobasket_editions(season = 2025)
fiba_weurobasket_schedule(season = 2025)
```

## Value

A `wehoop_data` tibble with the columns of the base wrapper (the raw JSON list when `return_parsed = FALSE`), one table per function:

`fiba_weurobasket_editions()`:

The columns of `fiba_gdap_editions()`:

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |

| | | |
|---|---|---|
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_weurobasket_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is know |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |

| | | |
|---|---|---|
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gan |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_weurobasket_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |

`fiba_weurobasket_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |

| | | |
|---|---|---|
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_weurobasket_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |

| | | |
|---|---|---|
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_weurobasket_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |

| | | |
|---|---|---|
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_weurobasket_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not rec |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |

| | | |
|---|---|---|
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_weurobasket_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB tean |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: `fiba_ecw_competition`, `fiba_elw_competition`, `fiba_wafrobasket_competition`, `fiba_wamericup_competition`, `fiba_wasiacup_competition`, `fiba_woly_competition`, `fiba_wwc_competition`

---

# `fiba_woly_competition` FIBA competition: Women's Olympic Basketball Tournament

---

## Description

Per-competition aliases of the FIBA wrappers for Women's Olympic Basketball Tournament (GDAP competition family 352), named as in sdv-py. Season rule: event year (Paris 2024 = 2024).

## Usage

```
fiba_woly_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_woly_schedule(season = NULL, comp = NULL, return_parsed = TRUE)
```

```
fiba_woly_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_woly_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_woly_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_woly_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_woly_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_woly_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (Paris 2024 = 2024)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_woly_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_woly_editions() only). |
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See fiba_gdap_awards() for the shared notes.

A comp of another competition family, or one GDAP gives no family, is wehoop_asset_fetch; a season with no edition is wehoop_no_data; a season listing several editions, or a game function called with neither season nor comp, is wehoop_bad_request.

```
fiba_woly_editions(season = 2024)
fiba_woly_schedule(season = 2024)
```

## Value

A wehoop_data tibble with the columns of the base wrapper (the raw JSON list when return_parsed = FALSE), one table per function:

fiba_woly_editions():

The columns of fiba_gdap_editions():

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_woly_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |

| | | |
|---|---|---|
| team_b_official_name | character | Official name of team B: the cour |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is known |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairin |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gam |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |

| | | |
|---|---|---|
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_woly_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a natio |

`fiba_woly_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |

| | | |
|---|---|---|
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |

`fiba_woly_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |

| | | |
|---|---|---|
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_woly_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |

| | | |
|---|---|---|
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_woly_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not re |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_woly_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |

| | | |
|---|---|---|
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: fiba_ecw_competition, fiba_elw_competition, fiba_wafrobasket_competition, fiba_wamericup_competition, fiba_wasiacup_competition, fiba_weurobasket_competition, fiba_wwc_competition

---

# fiba_wwc_competition FIBA competition: FIBA Women's Basketball World Cup

---

## Description

Per-competition aliases of the FIBA wrappers for FIBA Women's Basketball World Cup (GDAP competition family 306), named as in sdv-py. Season rule: event year (2022 = 2022).

## Usage

```
fiba_wwc_editions(season = NULL, to_season = NULL, return_parsed = TRUE)

fiba_wwc_schedule(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_standings(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_teams(season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_player_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_team_box(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_pbp(game_id, season = NULL, comp = NULL, return_parsed = TRUE)

fiba_wwc_shots(game_id, season = NULL, comp = NULL, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| game_id | FIBA game id (game_id in the schedule). |
| season | Edition season key (event year (2022 = 2022)). Omitted = the newest edition that has started; a game function needs season or comp. For fiba_wwc_editions(): the first season (inclusive), omitted = every edition. |
| to_season | Last season, inclusive (fiba_wwc_editions() only). |

| | |
|---|---|
| comp | FIBA competition id (one per edition); skips the edition lookup and is needed when a season lists several editions. Checked to be an edition of this competition (one cached getGdapCompetitionById request). |
| return_parsed | Parse the payload into a tibble (default TRUE); FALSE returns the raw JSON list. |

## Details

FIBA's terms restrict this source: the GDAP developer-portal terms limit API use to 'Authorized Users' and the fiba.basketball terms say no portion of the site may be duplicated, redistributed or manipulated. The package owner reviewed both clauses on 2026-10-12, accepted them and approved the full intake, published data releases included. Users who are not themselves authorised by FIBA should check the terms before using the wrappers. See fiba_gdap_awards() for the shared notes.

A comp of another competition family, or one GDAP gives no family, is wehoop_asset_fetch; a season with no edition is wehoop_no_data; a season listing several editions, or a game function called with neither season nor comp, is wehoop_bad_request.

```
fiba_wwc_editions(season = 2022)
fiba_wwc_schedule(season = 2022)
```

## Value

A wehoop_data tibble with the columns of the base wrapper (the raw JSON list when return_parsed = FALSE), one table per function:

fiba_wwc_editions():

The columns of fiba_gdap_editions():

| col_name | type | description |
|---|---|---|
| hosts | character | Host countries with their cities. Stored as a JS |
| competition_id | integer | FIBA competition id (GDAP key space): one p |
| competition_code | character | Competition code (WMM men's World Cup, W |
| season | integer | Season key, three rules: the END year for club |
| official_name | character | Edition title (FIBA Basketball World Cup; Pari |
| start | character | First day, YYYY-MM-DD. |
| end | character | Last day, YYYY-MM-DD. |
| status | character | Competition status: In Progress, Completed, C |
| age_category | character | Age category (Senior, Under 16...). |
| gender | character | Men or Women: the gender of the competition |
| fiba_zone | character | FIBA zone the competition belongs to (World, |
| zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Europe; zone ta |
| zone_information_zone_official_name | character | Official name of the competition's FIBA zone ( |
| zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, OC). |
| zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| zone_information_top_zone_official_name | character | Parent zone name; always null in the samples, |
| zone_information_top_zone_code | character | Parent zone code; always null in the samples, v |
| competition_type | character | National Teams or Clubs. |
| competition_category_is_league | logical | True for league-format club competitions. |

| | | |
|---|---|---|
| competition_category_age_type | character | Age type code: S senior, U16. |
| competition_category_code | character | Category code (WC World Cup, OLY Olympic |
| competition_category_name | character | Name of the competition's category (FIBA Bas |
| official_flag_item_code | character | Official-status code: O = official. |
| official_flag_item_name | character | Official-status label. |
| official | character | Official-status label. |
| discipline | character | Discipline (Basketball; 3x3 is a separate platfo |
| competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = World Cup, 112 |
| competition_marketing_name_marketing_name | character | Competition family name; slugified it is the <h |
| competition_marketing_name_code | character | Family code; null in every sample. |

`fiba_wwc_schedule()`:

The columns of `fiba_gdap_schedule()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id, global across com |
| game_name | character | Technical game name '<roundId: |
| game_number | character | Game number within the round o |
| status_code | character | Game status code: VALID (valida |
| team_a_team_id | integer | Competition team id: one per tea |
| team_a_organisation_id | integer | Organisation id, stable across con |
| team_a_code | character | Team code (national: FIBA count |
| team_a_official_name | character | Official name of team A: the cou |
| team_a_short_name | character | Short name of team A: the countr |
| team_b_team_id | integer | Competition team id: one per tea |
| team_b_organisation_id | integer | Organisation id, stable across con |
| team_b_code | character | Team code (national: FIBA count |
| team_b_official_name | character | Official name of team B: the cou |
| team_b_short_name | character | Short name of team B: the countr |
| team_a_score | integer | Final score of team A. |
| team_b_score | integer | Final score of team B. |
| is_live | logical | True while the game is being play |
| current_period | character | Last period reached ('Q4', 'OT1' |
| chrono | character | Game clock of the current period, |
| live_game_status | integer | Live state code: 999 on every fini |
| current_period_status | character | State of the current period: E = en |
| host_city | character | City of the venue. |
| host_country | character | Country of the venue (name; a FI |
| host_country_code | character | FIBA code of the venue's country |
| venue_id | integer | Venue id (GDAP key space; 0 = u |
| venue_name | character | Name of the arena the game is pla |
| game_date_time | character | Tip-off in venue-local wall-clock |
| game_date_time_utc | character | Tip-off in UTC, ISO 8601 withou |
| has_time_game_date_time | logical | False when only the date is know |
| iana_time_zone | character | IANA time zone of the venue. |
| utc_offset | double | Venue UTC offset at tip-off, in ho |
| is_postponed | logical | True when the game was postpon |

| | | |
|---|---|---|
| is_played_behind_closed_doors | logical | True when played without spectat |
| venue_capacity | integer | Venue capacity (0 = unknown). |
| spectators | integer | Attendance at the game, as record |
| statistic_system | character | Live-stats system that scored the g |
| group_id | character | Group id (GDAP key space). |
| group_pairing_code | character | Group letter (group play) or pairi |
| round_round_id | integer | Round id (GDAP key space, uniq |
| round_round_code | character | Round code (GP group phase, QF |
| round_round_name | character | Round label (Group Phase, Quart |
| round_round_number | integer | Round number within the compet |
| round_round_type | character | Round type label: Group Round, |
| round_round_status_code | character | Round status: COMPL completed |
| round_game_system | character | Pairing system label (Single, Hon |
| round_game_system_code | character | Pairing system code: S single gam |
| round_start_date | character | Round start, local date-time. |
| round_end_date | character | Round end, local date-time (23:59 |
| round_round_kind | character | Standard Round or Qualifiers. |
| round_round_order | integer | Display order of the round. |
| competition_competition_id | integer | FIBA competition id (GDAP key |
| competition_competition_code | character | Competition code (WMM men's |
| competition_season | integer | Season key, three rules: the END |
| competition_official_name | character | Edition title (FIBA Basketball Wo |
| competition_start | character | First day, YYYY-MM-DD. |
| competition_end | character | Last day, YYYY-MM-DD. |
| competition_status | character | Competition status: In Progress, C |
| competition_age_category | character | Age category (Senior, Under 16... |
| competition_gender | character | Men or Women: the gender of the |
| competition_fiba_zone | character | FIBA zone the competition belon |
| competition_zone_information_zone_id | integer | FIBA zone id (1 = World, 23 = Eu |
| competition_zone_information_zone_official_name | character | Official name of the competition's |
| competition_zone_information_zone_code | character | Zone code (W, E, AF, AM, AS, O |
| competition_zone_information_top_zone_id | integer | Parent zone id; null for top zones. |
| competition_zone_information_top_zone_official_name | character | Parent zone name; always null in |
| competition_zone_information_top_zone_code | character | Parent zone code; always null in t |
| competition_competition_type | character | National Teams or Clubs. |
| competition_competition_category_is_league | logical | True for league-format club comp |
| competition_competition_category_age_type | character | Age type code: S senior, U16. |
| competition_competition_category_code | character | Category code (WC World Cup, C |
| competition_competition_category_name | character | Name of the competition's catego |
| competition_official_flag_item_code | character | Official-status code: O = official. |
| competition_official_flag_item_name | character | Official-status label. |
| competition_official | character | Official-status label. |
| competition_discipline | character | Discipline (Basketball; 3x3 is a se |
| competition_competition_marketing_name_competition_marketing_name_id | integer | Competition family id (201 = Wo |
| competition_competition_marketing_name_marketing_name | character | Competition family name; slugifi |
| competition_competition_marketing_name_code | character | Family code; null in every sample |
| game_statistic_status_code | character | Box-score validation status: VAL |
| game_play_by_play_status_code | character | Play-by-play availability flag: VA |

| | | |
|---|---|---|
| game_result_status_code | character | Result validation status: VALID. |
| game_result_additional_information | character | Free-text result note (forfeit, over |
| window_code | character | Qualifier window code; null other |
| window_name | character | Qualifier window name; null othe |
| team_a_from | character | How team A qualified for this slo |
| team_b_from | character | How team B qualified for this slot |

`fiba_wwc_standings()`:

The columns of `fiba_gdap_final_standings()`:

| col_name | type | description |
|---|---|---|
| competition_id | integer | FIBA competition id (Int64) the final ranking belongs to; the comp argument. |
| final_rank | integer | Final rank in the competition. |
| team_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at the 2023 Worl |
| team_team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| team_short_name | character | Short team name: the country code for a national team, the club's short name for a club. |
| team_profile | character | Team profile; null in this route. |
| team_status_code | character | Team entry status; always null in the samples, values unverified. |
| team_status_name | character | Team entry status label; always null in the samples, values unverified. |
| team_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federation for a nati |

`fiba_wwc_teams()`:

The columns of `fiba_gdap_teams()`:

| col_name | type | description |
|---|---|---|
| team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| team_code | character | Team abbreviation (FIBA country code or club abbreviation). |
| short_name | character | Short team name: the country code for a national team, the club's short name |
| profile_team_id | integer | Competition team id: one per team per competition (Germany = 279569 at th |
| profile_organisation_id | integer | Organisation id, stable across competitions and seasons: the national federati |
| profile_name | character | Team name as registered for the competition. |
| profile_code | character | Team code (FIBA country code or club abbreviation). |
| profile_country_name | character | Country of the team, in English. |
| profile_country_code | character | Country FIBA code. |
| profile_gender | character | Men or Women: the gender of the team's competition. |
| profile_facebook | character | Facebook handle or URL; always null in the samples, values unverified. |
| profile_twitter | character | X/Twitter handle; always null in the samples, values unverified. |
| profile_instagram | character | Instagram handle; always null in the samples, values unverified. |
| profile_website | character | The team's website address. |
| profile_you_tube | character | YouTube handle; always null in the samples, values unverified. |
| profile_final_ranking | integer | Final rank in the competition. |
| profile_points_per_game | double | The team's average points per game played. |
| profile_rebounds_per_game | double | The team's average rebounds per game played. |
| profile_assists_per_game | double | The team's average assists per game played. |
| profile_rank_points_per_game | double | Rank in the competition by points per game (1 = highest). |
| profile_rank_rebounds_per_game | double | Rank in the competition by rebounds per game (1 = highest). |

| | | |
|---|---|---|
| profile_rank_assists_per_game | double | Rank in the competition by assists per game (1 = highest). |
| profile_total_games_won | integer | The team's total games won over the games played. |
| profile_total_games_lost | integer | The team's total games lost over the games played. |
| profile_social_networks | character | Social-network handles. Stored as a JSON string. |
| profile_competition_id | integer | FIBA competition id (GDAP key space): one per edition (2023 World Cup = |
| status_code | character | Team entry status: APPR approved. |
| status_name | character | Team entry status label (Approved). |
| organisation_id | integer | Organisation id, stable across competitions and seasons: the national federatio |

`fiba_wwc_player_box()`:

The columns of `fiba_gamepage_box_players()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-p |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not sa |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germa |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-pl |
| first_name | character | Given name(s) as registered with FIBA. |
| last_name | character | Last name as registered with FIBA. |
| jersey | character | Jersey number as a string. |
| position | character | Playing position code from the lineup card (PG, SG, SF, PF, C). |
| is_captain | logical | True for the team captain. |
| is_starter | logical | True for the starting five. |
| has_played | logical | True when the player got on the court. |
| minutes | character | Minutes played as an 'm:ss' string ('0:0' = did not play). |
| seconds_played | integer | Seconds played, derived from minutes. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |

| | | |
|---|---|---|
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sampled |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goals - m |
| plus_minus | integer | Plus/minus: the team's point differential while the player was on the court. |
| cb | logical | Boolean flag, false on every sampled player; meaning unverified. |

`fiba_wwc_team_box()`:

The columns of `fiba_gamepage_box_teams()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a ga |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competition |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does n |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = G |
| score | integer | Team final score. |
| points | integer | Points scored in the game. |
| fg_made | integer | Field goals made (2 + 3 points). |
| fg_attempted | integer | Field goals attempted (2 + 3 points). |
| fg_pct | double | Field-goal percentage, 0-100. |
| fg2_made | integer | Two-point field goals made. |
| fg2_attempted | integer | Two-point field goals attempted. |
| fg2_pct | double | Two-point field-goal percentage, 0-100. |
| fg3_made | integer | Three-point field goals made. |
| fg3_attempted | integer | Three-point field goals attempted. |
| fg3_pct | double | Three-point field-goal percentage, 0-100. |
| ft_made | integer | Free throws made. |
| ft_attempted | integer | Free throws attempted. |
| ft_pct | double | Free-throw percentage, 0-100. |
| reb_offensive | integer | Offensive rebounds. |
| reb_defensive | integer | Defensive rebounds. |
| reb_total | integer | Total rebounds (a team row includes the team rebounds team_reb_total). |
| assists | integer | Assists credited in the game. |
| steals | integer | Steals credited in the game. |
| blocks | integer | Blocked shots (blocks made). |
| blocks_received | integer | Own shots blocked by the opponent (blocks received). |
| turnovers | integer | Turnovers (a team row includes the team turnovers team_turnovers). |
| fouls | integer | Personal fouls (a team row includes the team fouls team_fouls). |
| fouls_drawn | integer | Fouls drawn in the game. |
| paint_fg_made | integer | Field goals made in the paint (team points_in_paint = 2 x paint_fg_made on every sam |
| paint_fg_attempted | integer | Field goals attempted in the paint. |
| paint_fg_pct | double | Field-goal percentage in the paint, 0-100. |
| efficiency | integer | Efficiency (FIBA EFF: points + rebounds + assists + steals + blocks - missed field goal |

| | | |
|---|---|---|
| biggest_lead | integer | Biggest lead, points. |
| biggest_lead_score | character | Score when the biggest lead was reached, 'A-B' string (spacing varies; empty when the |
| biggest_run | integer | Biggest scoring run, points. |
| biggest_run_score | character | Score of the biggest scoring run, 'x-y' string; exact meaning unverified. |
| fast_break_points | integer | Fast-break points. |
| points_off_turnovers | integer | Points scored after turnovers. |
| bench_points | integer | Points from the bench (the sum of the non-starters' points). |
| points_in_paint | integer | Points in the paint (2 x paint_fg_made). |
| second_chance_points | integer | Second-chance points. |
| a_act | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_ACT). |
| a_foul | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_FOUL). |
| a_tod | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOD). |
| a_tot | integer | Always 0 in the samples; meaning unverified (FIBA stat code A_TOT). |
| team_reb_offensive | integer | Team offensive rebounds, credited to no player. |
| team_reb_defensive | integer | Team defensive rebounds, credited to no player. |
| team_reb_total | integer | Team rebounds (total rebounds minus the players' sum). |
| team_fouls | integer | Team (bench/coach) fouls (personal fouls minus the players' sum). |
| team_turnovers | integer | Team turnovers (turnovers minus the players' sum). |
| team_bench_points_flag | integer | Bench points in FLS games (equal to bench_points); 0 in OVR games (FIBA stat code |
| time_led | character | Time the team led, 'mm:ss' string. |
| lead_changes | integer | Lead changes in the game (game-level; repeats on both team rows). |
| times_tied | integer | Times the score was tied (game-level; repeats on both team rows). |
| duration | character | Game duration as an 'h:mm' string (game-level; repeats on both team rows). |
| attendance | integer | Attendance (game-level; repeats on both team rows). |

`fiba_wwc_pbp()`:

The columns of `fiba_gamepage_pbp()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| wall_time_ms | double | Wall-clock time of the action, Unix epoch milliseconds. (Int64 in sdv-py; numeric here, the va |
| score_a | integer | Team A score after the action. |
| score_b | integer | Team B score after the action. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| action_family | character | Action family: shot, subst, timeout, periods, and 'unknwown' (FIBA's spelling) for every othe |
| description | character | English action text ('3pt jump shot missed', 'defensive rebound'); null on most OVR actions. |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |

| | | |
|---|---|---|
| player2_id | integer | FIBA person id (Int64) of the second player on a substitution; null when the game does not rec |
| sub_direction | character | Substitution direction: IN or OUT; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

`fiba_wwc_shots()`:

The columns of `fiba_gamepage_shots()`:

| col_name | type | description |
|---|---|---|
| game_id | integer | FIBA game id (Int64), global across competitions: gdapGameId, the <gameId> of a game-pag |
| competition_id | integer | FIBA competition id (Int64) of the game's edition; the comp argument. |
| statistic_system | character | Live-stats system of the game: OVR (World Cup, Olympics) or FLS (club competitions, most |
| period | character | Period name: Q1..Q4, then OT1, OT2... |
| period_number | integer | 1-based position of the period in the game. |
| action_id | character | Action id (String): '<n>:<seq>' in OVR games (3:0015), a sequence number in FLS games; u |
| order | integer | Sort key within the period (OVR: 1, 2, ...; FLS: 1000, 2000, ...); null on some period events. |
| clock | character | Game clock: time REMAINING in the period, 'm:ss', counting down from 10:00 (5:00 in ove |
| clock_seconds | integer | Seconds remaining in the period, derived from clock. |
| action_code | character | Action code: P2 two-point shot, P3 three-point shot, FT free throw, REB rebound, TREB team |
| organisation_id | integer | Stable federation or club id (Int64) of the team, the same in every competition (64 = Germany |
| team_side | character | A or B: whether the row is the schedule's team A or team B (null when the page does not say) |
| player_id | integer | FIBA person id (Int64), global across competitions and roles: the page's P_<id>, play-by-play |
| shot_type | character | Shot type derived from action_code: 2PT, 3PT or FT; null on other actions. |
| points_attempted | integer | Points the attempt was worth (2, 3 or 1), made or not; null on other actions. |
| made | logical | True when the shot went in (two-pointers, three-pointers and free throws only; null otherwise) |
| shot_x | integer | Shot x, integer pixels on a half-court image across the court width (observed 0-278, basket nea |
| shot_y | integer | Shot y, integer pixels from the baseline (y = 0) towards midcourt (observed 0-224, basket near |
| has_location | logical | True when shot_x / shot_y locate the shot (two- and three-pointers); False for free throws (alw |

## See Also

Other FIBA Competition Functions: `fiba_ecw_competition`, `fiba_elw_competition`, `fiba_wafrobasket_competition`, `fiba_wamericup_competition`, `fiba_wasiacup_competition`, `fiba_weurobasket_competition`, `fiba_woly_competition`

---

`fox_basketball_league_leaders`

**Get Fox Sports basketball statistical leaders**

---

## Description

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) statistical leaders.** `fox_wnba_league_leaders()` hits the wnba slug; `fox_wbb_league_leaders()` hits the wcbk slug.

## Usage

```
fox_wnba_league_leaders(category = "scoring", who = "player", page = 0)

fox_wbb_league_leaders(category = "scoring", who = "player", page = 0)
```

## Arguments

| | |
|---|---|
| `category` | Stat category (default `"scoring"`). |
| `who` | `"player"` or `"team"` (default `"player"`). |
| `page` | 0-based page index (default `0`). |

## Value

A `wehoop_data` tibble of leaderboard rows (`entity_id` + stat columns).

## See Also

Other Fox Sports Functions: `fox_wbb_teams_all()`, `fox_wnba_pbp()`, `fox_wnba_team_roster()`

## Examples

```
try(fox_wnba_league_leaders("scoring"))


try(fox_wbb_league_leaders("scoring"))
```

---

# fox_wbb_teams_all — Get the full Fox Sports women's college basketball team directory

---

## Description

**Enumerate every WBB team in the Fox Sports (Bifrost) directory.** A single `fox_wbb_teams()` call only returns the seed team's conference, so this walks unseen team ids (calling one seed per conference) and unions the results.

## Usage

```
fox_wbb_teams_all(max_id = 500, max_calls = 60)
```

### Arguments

| | |
|---|---|
| max_id | Highest Fox team id to probe as a seed (default 500). |
| max_calls | Safety cap on the number of standings calls (default 60). |

### Value

A wehoop_data tibble, one row per team: fox_team_id, fox_team_name, fox_section.

### See Also

Other Fox Sports Functions: fox_basketball_league_leaders, fox_wnba_pbp(), fox_wnba_team_roster()

### Examples

```
try(fox_wbb_teams_all())
```

---

## fox_wnba_pbp — Get Fox Sports basketball play-by-play

---

### Description

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) play-by-play.** fox_wnba_pbp() hits the wnba slug; fox_wbb_pbp() hits the wcbk slug.

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) boxscore.** fox_wnba_boxscore() hits the wnba slug; fox_wbb_boxscore() hits the wcbk slug.

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) game odds.** fox_wnba_odds() hits the wnba slug; fox_wbb_odds() hits the wcbk slug.

### Usage

```
fox_wnba_pbp(game_id)

fox_wbb_pbp(game_id)

fox_wnba_boxscore(game_id)

fox_wbb_boxscore(game_id)

fox_wnba_odds(game_id)

fox_wbb_odds(game_id)
```

### Arguments

| | |
|---|---|
| game_id | Fox Bifrost event id. |

## Value

A wehoop_data tibble, one row per play: game_id, period, left_team, right_team, play_id, clock, team, left_score_change, right_score_change, play_text.

A wehoop_data tibble (long), one row per (player, stat): game_id, team, stat_group, player, athlete_id, stat, value.

A wehoop_data tibble, one row per team: game_id, team, plus the six-pack odds columns (spread / to-win / total). Empty when no market.

## See Also

Other Fox Sports Functions: fox_basketball_league_leaders, fox_wbb_teams_all(), fox_wnba_team_roster()

## Examples

```
try(fox_wnba_pbp("2215"))


try(fox_wbb_pbp("388986"))


try(fox_wnba_boxscore("2215"))


try(fox_wbb_boxscore("388986"))


try(fox_wnba_odds("2215"))


try(fox_wbb_odds("388986"))
```

---

# fox_wnba_team_roster Get Fox Sports basketball team roster

---

## Description

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) team roster.** fox_wnba_team_roster() hits the wnba slug; fox_wbb_team_roster() hits the wcbk slug.

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) team stat leaders.** fox_wnba_team_stats() hits the wnba slug; fox_wbb_team_stats() hits the wcbk slug.

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) team game log.** fox_wnba_team_gamelog() hits the wnba slug; fox_wbb_team_gamelog() hits the wcbk slug.

**Get Fox Sports (Bifrost) WNBA / women's college basketball (WBB) standings.** fox_wnba_standings() hits the wnba slug; fox_wbb_standings() hits the wcbk slug.

**Get the Fox Sports (Bifrost) WNBA / women's college basketball (WBB) team directory**, derived from the standings endpoint. fox_wnba_teams() hits the wnba slug; fox_wbb_teams() hits the wcbk slug.

## Usage

```
fox_wnba_team_roster(team_id)

fox_wbb_team_roster(team_id)

fox_wnba_team_stats(team_id)

fox_wbb_team_stats(team_id)

fox_wnba_team_gamelog(team_id)

fox_wbb_team_gamelog(team_id)

fox_wnba_standings(team_id)

fox_wbb_standings(team_id)

fox_wnba_teams(team_id = "3")

fox_wbb_teams(team_id = "11")
```

## Arguments

| | |
|---|---|
| team_id | Fox Bifrost seed team id used to fetch league standings (default "3"). The standings response enumerates every team in the seed's league sections. |

## Value

A wehoop_data tibble, one row per player: team_id, position_group, player, position/age/etc. columns, athlete_id.

A wehoop_data tibble: team_id, category, stat, stat_abbreviation, player, value.

A wehoop_data tibble (long): team_id, season_type, category, game_id, game_date, opponent, stat, value.

A wehoop_data tibble of standings rows (team_id, section, the standings columns, entity_id).

A wehoop_data tibble, one row per team: fox_team_id, fox_team_name, fox_section.

## See Also

Other Fox Sports Functions: fox_basketball_league_leaders, fox_wbb_teams_all(), fox_wnba_pbp()

## Examples

```
try(fox_wnba_team_roster("1"))


try(fox_wbb_team_roster("11"))
```

```
try(fox_wnba_team_stats("1"))

try(fox_wbb_team_stats("11"))

try(fox_wnba_team_gamelog("1"))

try(fox_wbb_team_gamelog("11"))

try(fox_wnba_standings("1"))

try(fox_wbb_standings("11"))

try(fox_wnba_teams())

try(fox_wbb_teams("11"))
```

---

`genius_competition_games`

## Genius Competition Games

---

### Description

The recent-games window of a competition, with the global Genius team ids.

Endpoint: `GET https://fibalivestats.dcd.shared.geniussports.com/data/competition/{comp_id}.json`

A few recent or upcoming games, not the season. Scores are cast to integer (NA before tip-off). `match_time` is local (zone not stated), `match_time_utc` UTC.

### Usage

```
genius_competition_games(comp_id, return_parsed = TRUE)
```

### Arguments

| | |
|---|---|
| `comp_id` | Genius competition id: one per org competition-season (`comp_id` of `genius_competitions()`), e.g. 49915. |
| `return_parsed` | Parse into a tibble (default `TRUE`); `FALSE` returns the parsed body or page record (a list): `data.json` as served plus `_request` (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the `GENIUS_PROXY` environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json` carries no global team or player id: `tno` (1 or 2) and `pno` are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when Genius has nothing for the request (HTTP 404, the S3 `AccessDenied` 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); `wehoop_bad_request` for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); `wehoop_asset_fetch` when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| league_id | integer | Genius league id of the org (ESAKE = 115); the leagueId column of the hosted choos |
| competition_id | integer | Genius competition id (compId): one per org competition-season; the {compId} of ho |
| match_id | integer | Genius match id: the {matchId} of data.json and /u/; one global sequence across ever |
| match_status | character | Match status: COMPLETE, SCHEDULED, IN_PROGRESS, ... (upper case) |
| match_time | character | Local tip-off time, 'YYYY-MM-DD HH:MM:SS' (venue time zone, not stated) |
| match_time_utc | character | Tip-off time in UTC, 'YYYY-MM-DD HH:MM:SS' |
| live | integer | 1 while the game is live, else 0 |
| period | integer | Current period of a live or finished game (present on some rows only) |
| period_type | character | Current period type: REGULAR or OVERTIME (present on some rows only) |
| clock | character | Period clock of a live or finished game, 'mm:ss' remaining (present on some rows on |
| home_team_id | integer | Home team: Genius team id (warehouse key space, global across orgs and seasons; th |
| home_name | character | Home team full name, in the competition's own script when the org enters one. |
| home_name_international | character | Home team name in Latin script |
| home_code | character | Home team short code, in the competition's own script when the org enters one. |
| home_score | integer | Home points: a string on most rows, an integer on some (" before tip-off); cast it |
| away_team_id | integer | Away team: Genius team id (warehouse key space, global across orgs and seasons; th |
| away_name | character | Away team full name, in the competition's own script when the org enters one. |
| away_name_international | character | Away team name in Latin script |

| | | |
|---|---|---|
| away_code | character | Away team short code, in the competition's own script when the org enters one. |
| away_score | integer | Away points: a string on most rows, an integer on some (" before tip-off); cast it |

## See Also

Other Genius Functions: genius_competitions(), genius_game(), genius_game_crosswalk(), genius_match_competition(), genius_pbp(), genius_player_box(), genius_schedule(), genius_schedule_embed( genius_shots(), genius_team_box(), genius_team_crosswalk(), genius_widget_games(), lfb_competition, wbbl_competition

## Examples

```
try(genius_competition_games(comp_id = 49915))
```

---

# genius_competitions Genius Competitions

---

## Description

Every competition of an org from the hosted WebStats chooser, with its END-year season.

Endpoint: `GET https://hosted.dcd.shared.geniussports.com/{org}/en/schedule`

One row per competition-season. `year` is the Genius year (START year for a two-year org, the year itself for a calendar org) and `season` the sdv END year: a two-year span in the label wins (`2019 - 20` is 2020), else the Genius year plus one for a split org, else the year itself; `season_rule` names the rule (NA season when none applies). `selected` marks the competition the chooser pre-selects.

## Usage

```
genius_competitions(org, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| org | Genius ORG code of the league (ESAKE, CEBL, BNXT, SLB, ...); see genius_competitions(). |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the `GENIUS_PROXY` environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json` carries no global team or player id: `tno` (1 or 2) and `pno` are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when Genius has nothing for the request (HTTP 404, the S3 `AccessDenied` 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); `wehoop_bad_request` for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); `wehoop_asset_fetch` when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| org | character | ORG code the request used (the hosted ORG, or the LiveStats ORG for the /u/ shell). |
| comp_id | integer | Genius competition id (compId): one per org competition-season; the {compId} of hosted and data/ |
| label | character | Competition label as the org typed it, free text in any language: may or may not carry a season ('BA |
| league_id | integer | Genius league id of the org (ESAKE = 115; IBBA has two: 187 and 549) |
| year | integer | Genius season year of the competition: the START year of a two-year season (BNXT '2023-2024' = |
| season | integer | sdv END-year season of the competition (2026 = 2025-26) from the label span, else the Genius yea |
| season_rule | character | Which rule produced season: 'label span', 'genius year + 1 (split)', 'genius year (calendar)', 'label |
| selected | logical | True for the competition the hosted chooser pre-selects (the org's default, not always the newest). |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed(`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

## Examples

```
try(genius_competitions(org = "ESAKE"))
```

genius_game **Genius Game**

## Description

One row per finished game: period and clock state, both teams' names, codes and scores, officials.

Endpoint: GET https://fibalivestats.dcd.shared.geniussports.com/data/{match_id}/data.json

team1 is the team listed first. referee1 ... commissioner read officials.<role> (current feeds) or the flat officials_refereeN strings (older feeds); NA when absent.

## Usage

```
genius_game(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| match_id | Genius match id (match_id of genius_schedule(), genius_widget_games() or genius_competition_games()), e.g. 2902690. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set options(wehoop.proxy = ...) or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (comp_id): genius_competitions() lists an org's competitions with the sdv END-year season (a START-year org lists 2025-26 as season 2026), genius_match_competition() maps a game to its competition, and data.json itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of /u/ (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. data.json carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (transient = TRUE: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for, st |
| period | integer | Current (or last) period number; restarts at 1 in overtime |
| period_type | character | Current period type: REGULAR or OVERTIME |
| in_ot | integer | 1 when the game is (or ended) in overtime, else 0 |
| clock | character | Clock of the current period at capture time, 'mm:ss' remaining ('00:00' once the game |
| periods_max | integer | Number of regulation periods (4) |
| period_length_regular | integer | Length of a regulation period, minutes (10 under FIBA rules, 12 in some leagues) |
| period_length_overtime | integer | Length of an overtime period, minutes |
| attendance | integer | Attendance as entered by the org; 0 when not entered |
| disable_match | integer | 1 when the org disabled the public feed for this game, else 0 |
| total_time_added | integer | Clock time added by corrections, seconds (unverified; 0 in every sample) |
| team1_name | character | Full name of team 1, the team listed first (the home team on the hosted schedule and wi |
| team1_code | character | Short code of team 1, in the competition's own script when the org enters one. |
| team1_score | integer | Final points of team 1 (cast to an integer; the 2014-15 feed sends a string). |
| team2_name | character | Full name of team 2, the team listed second. |
| team2_code | character | Short code of team 2, in the competition's own script when the org enters one. |
| team2_score | integer | Final points of team 2 (cast to an integer; the 2014-15 feed sends a string). |
| referee1 | character | First referee's name, from officials.referee1 (international form) or the flat officials_refe |
| referee2 | character | Second referee's name, from officials.referee2 (international form) or the flat officials_r |
| referee3 | character | Third referee's name when the feed lists one; null otherwise. |
| commissioner | character | Commissioner (game supervisor) name when the feed lists one; null otherwise. |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed(`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

## Examples

```
try(genius_game(match_id = 2902690))
```

---

`genius_game_crosswalk` **Genius Game Crosswalk**

---

## Description

Provider game key <-> Genius LiveStats match id, from a frame that carries both.

Reads a frame you already have (no request is made), checks the id types and returns the distinct id pairs, one-to-one both ways. Rows without a Genius id (older seasons, upcoming games) are dropped.

## Usage

```
genius_game_crosswalk(df, source)
```

## Arguments

| | |
|---|---|
| df | A data frame from the wrapper named for source. |
| source | One of "sportpress", "cebl", "tbf", "kzs", "sportality". |

## Details

Sources: sportpress (BNXT sportpress_game_slider: game_id / genius_id), cebl (game_id / genius_match_id), tbf (match_id / genius_id), kzs (match_id / genius_match_id) and sportality (game_uuid / game_id, the Genius match id). The provider key is integer for every source but sportality (a character uuid); the Genius id is integer. A wrong source, a missing column, a wrong id type or a key that maps to two ids raises wehoop_bad_request.

## Value

A wehoop_data tibble, one row per game:

| col_name | type | description |
|---|---|---|
| game_key | integer or character | Provider's own game key: the BNXT, CEBL, TBF or KZS integer game id, or the S |
| genius_match_id | integer | Genius Sports LiveStats match id of the same game (the match_id every genius_* w |

## See Also

Other Genius Functions: genius_competition_games(), genius_competitions(), genius_game(), genius_match_competition(), genius_pbp(), genius_player_box(), genius_schedule(), genius_schedule_embed( genius_shots(), genius_team_box(), genius_team_crosswalk(), genius_widget_games(), lfb_competition, wbbl_competition

## Examples

```
slider <- data.frame(game_id = c(1L, 2L), genius_id = c(2902690L, NA))
genius_game_crosswalk(slider, "sportpress")
```

genius_match_competition

## Genius Match Competition

### Description

The competition (comp_id and name) a game belongs to, from its LiveStats HTML shell.

Endpoint: GET https://fibalivestats.dcd.shared.geniussports.com/u/{org}/{match_id}/

A one-row tibble: the only way to map a data.json game to its competition (comp_id) and, through genius_competitions(), to a season. org is the ORG the request used.

### Usage

```
genius_match_competition(org, match_id, return_parsed = TRUE)
```

### Arguments

| | |
|---|---|
| org | LiveStats ORG of the game's shell (can differ from the hosted ORG: NZN vs NZNBL). |
| match_id | Genius match id (match_id of genius_schedule(), genius_widget_games() or genius_competition_games()), e.g. 2902690. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

### Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set options(wehoop.proxy = ...) or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (comp_id): genius_competitions() lists an org's competitions with the sdv END-year season (a START-year org lists 2025-26 as season 2026), genius_match_competition() maps a game to its competition, and data.json itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of /u/ (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. data.json carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch

when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

```
genius_match_competition(org = "ESAKE", match_id = 2902690)
```

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Match id from the page's hidden input #matchId (equals the {matchId} asked for). Genius m |
| comp_id | integer | Competition id from the hidden input #compId. Genius competition id (compId): one per org |
| competition_name | character | Competition label from #competitionName, free text ('Stoiximan GBL 2026-2027'); carries |
| org_label | character | First item of og:description: the org's display code (ESAKE) |
| og_description | character | The page's og:description: <ORG> , <competition>, Check out the full statistics |
| org | character | ORG code the request used (the hosted ORG, or the LiveStats ORG for the /u/ shell). |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

---

`genius_pbp` **Genius Play-by-Play**

---

## Description

Play-by-play of a finished game, newest action first, with the running score and lead.

Endpoint: `GET https://fibalivestats.dcd.shared.geniussports.com/data/{match_id}/data.json`

Rows in the feed's order: newest first by game time, so `row_index` 0 is the last action (normally the `game end` row); sort on `action_number` for entry order. `score_1` / `score_2` are always integer. `qualifier` joins the qualifier codes with commas. `clock`, `action_number` and `previous_action` are NA in the 2014-15 feed. Overtime restarts `period` at 1: read it with `period_type`.

## Usage

```
genius_pbp(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| match_id | Genius match id (match_id of genius_schedule(), genius_widget_games() or genius_competition_games()), e.g. 2902690. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set options(wehoop.proxy = ...) or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (comp_id): genius_competitions() lists an org's competitions with the sdv END-year season (a START-year org lists 2025-26 as season 2026), genius_match_competition() maps a game to its competition, and data.json itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of /u/ (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. data.json carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (transient = TRUE: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A wehoop_data tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when return_parsed = FALSE:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| row_index | integer | Position of the action in the feed order, 0 = the newest action by game time (normall |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted) |
| previous_action | integer | action_number of the linked earlier action (the shot an assist or rebound belongs to); |
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| gt | character | Game clock, 'mm:ss' remaining in the period |
| clock | character | Game clock with hundredths, 'mm:ss:cc' remaining in the period; absent in older fe |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |

| | | |
|---|---|---|
| player | character | Player display name; ” when no player |
| shirt_number | character | Player shirt number (string); ” when no player |
| first_name | character | Given name in the competition’s own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition’s own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name (’T. Nichols’), in the local script when the org enters one |
| action_type | character | Action type: 2pt, 3pt, freethrow, rebound, assist, turnover, steal, block, foul, foulon ( |
| sub_type | character | Action subtype, free-text code per actionType: shot type for 2pt/3pt (layup, drivingl |
| qualifier | character | Qualifier codes of the action joined with commas (pointsinthepaint, fromturnover, 2r |
| success | integer | 1 = made / successful, 0 = missed (shots, free throws); 1 for other actions |
| scoring | integer | 1 when the action is a scoring attempt (shot or free throw), else 0 |
| score_1 | integer | Team 1 score after the action, always an integer (the feed sends a string in most gam |
| score_2 | integer | Team 2 score after the action, always an integer (the feed sends a string in most gam |
| lead | integer | Score margin after the action, team 1 minus team 2 (negative when team 2 leads) |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

## Examples

```
try(genius_pbp(match_id = 2902690))
```

---

`genius_player_box` **Genius Player Box Score**

---

## Description

Player box score of a finished game: minutes, shooting, rebounds, assists and the efficiency formulas.

Endpoint: `GET https://fibalivestats.dcd.shared.geniussports.com/data/{match_id}/data.json`

Rows keyed by (`tno`, `pno`), both in-game numbers; `pno` is not the shirt number and the feed has no global player id (join across games on name, team code and `shirt_number`). Percentages are numeric 0-100; `minutes` is the printed `mm:ss` and `seconds_played` its seconds.

## Usage

```
genius_player_box(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| match_id | Genius match id (match_id of genius_schedule(), genius_widget_games() or genius_competition_games()), e.g. 2902690. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set options(wehoop.proxy = ...) or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (comp_id): genius_competitions() lists an org's competitions with the sdv END-year season (a START-year org lists 2025-26 as season 2026), genius_match_competition() maps a game to its competition, and data.json itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of /u/ (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. data.json carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (transient = TRUE: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A wehoop_data tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when return_parsed = FALSE:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |
| name | character | Display name ('T. Nichols') |
| first_name | character | Given name in the competition's own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition's own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name ('T. Nichols'), in the local script when the org enters one |
| shirt_number | character | Shirt number as a string ('0' and '00' differ) |

| | | |
|---|---|---|
| playing_position | character | Playing position code (PG, SG, SF, PF, C, G, F, ...); empty string when not entered |
| starter | integer | 1 when the player started the game, else 0 |
| captain | integer | 1 for the team captain, else 0; absent in some orgs |
| active | integer | 1 when the player is on court now (at game end: on court at the final whistle), else 0 |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '2 |
| field_goals_made | integer | Field goals the player made in the game (two- and three-pointers together; a count). |
| field_goals_attempted | integer | Field goals the player attempted in the game (two- and three-pointers together; a cou |
| field_goals_percentage | double | Field-goal percentage, 0-100, rounded to an integer by the feed |
| three_pointers_made | integer | Three-pointers made |
| three_pointers_attempted | integer | Three-pointers attempted |
| three_pointers_percentage | double | Three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Two-pointers made |
| two_pointers_attempted | integer | Two-pointers attempted |
| two_pointers_percentage | double | Two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Free throws the player made in the game (a count). |
| free_throws_attempted | integer | Free throws the player attempted in the game (a count). |
| free_throws_percentage | double | Free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Defensive rebounds |
| rebounds_offensive | integer | Offensive rebounds |
| rebounds_total | integer | Total rebounds (offensive plus defensive) the player collected in the game. |
| assists | integer | Assists credited to the player in the game. |
| turnovers | integer | Turnovers committed by the player in the game. |
| steals | integer | Steals credited to the player in the game. |
| blocks | integer | Shots the player blocked in the game (blocks made, not received). |
| blocks_received | integer | Own shots blocked by an opponent |
| fouls_personal | integer | Personal fouls committed |
| fouls_on | integer | Fouls drawn: personal fouls committed against the player in the game. |
| points | integer | Points the player scored in the game, free throws included. |
| points_from_turnovers | integer | Points off turnovers |
| points_second_chance | integer | Second-chance points |
| points_fast_break | integer | Fast-break points |
| points_in_the_paint | integer | Points the player scored in the paint in the game (the feed's sPointsInThePaint). |
| plus_minus_points | integer | Plus/minus: team point margin while the player was on court |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

## Examples

```
try(genius_player_box(match_id = 2902690))
```

---

`genius_schedule` **Genius Schedule**

---

## Description

Every match of a competition (hosted WebStats): ids, status, venue, teams and scores.

Endpoint: `GET https://hosted.dcd.shared.geniussports.com/{org}/en/competition/{comp_id}/schedule?roun`

One row per match in page order. Scores are NA before tip-off and for some completed games whose score the page does not print. `date_time` is the page's text (local time, zone not stated). `link_match_id` is NA for orgs that hide match pages (FFBB); `venue_id` is NA when the venue is plain text. An unfilled play-off slot ('To be determined', no team or match link) takes the scope the page's other blocks prove; a page of nothing but such slots is refused.

## Usage

```
genius_schedule(org, comp_id, round_number = -1, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `org` | Genius ORG code of the league (ESAKE, CEBL, BNXT, SLB, ...); see `genius_competitions()`. |
| `comp_id` | Genius competition id: one per org competition-season (`comp_id` of `genius_competitions()`), e.g. 49915. |
| `round_number` | Round to list; -1 (default) lists every round of the competition. |
| `return_parsed` | Parse into a tibble (default `TRUE`); `FALSE` returns the parsed body or page record (a list): `data.json` as served plus `_request` (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the `GENIUS_PROXY` environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json`

carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (transient = TRUE: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

## Value

A wehoop_data tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when return_parsed = FALSE:

| col_name | type | description |
|---|---|---|
| match_id | integer | From the block id extfix_<id>. Genius match id: the {matchId} of data.json and /u/; one gl |
| link_match_id | integer | Match id of the block's /match/<id>/summary link; equals match_id (the capture refuses a p |
| comp_id | integer | CompId of the match link, or of the team links when the org hides match pages (FFBB); equa |
| org | character | ORG code of the match link (or the team links, as comp_id) |
| status | character | Status from the block class STATUS_*: COMPLETE, SCHEDULED, POSTPONED, ... |
| date_time | character | Tip-off as displayed, 'Mon D, YYYY, h:mm AM' (local time, zone not stated); null when un |
| venue_id | integer | Genius venue id (the /venue/{id} of hosted pages); null when the venue is plain text (FFBB) |
| venue_name | character | Venue name as entered (local script) |
| home_team_id | integer | Home team: Genius team id (warehouse key space, global across orgs and seasons; the /team |
| home_team_name | character | Home team full name |
| home_team_code | character | Home team code (local script possible) |
| home_score | integer | Home points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |
| away_team_id | integer | Away team: Genius team id (warehouse key space, global across orgs and seasons; the /teamA |
| away_team_name | character | Away team full name |
| away_team_code | character | Away team code (local script possible) |
| away_score | integer | Away points; null before tip-off (typed integer only on a route whose 3 sampled rows were al |

## See Also

Other Genius Functions: genius_competition_games(), genius_competitions(), genius_game(), genius_game_crosswalk(), genius_match_competition(), genius_pbp(), genius_player_box(), genius_schedule_embed(), genius_shots(), genius_team_box(), genius_team_crosswalk(), genius_widget_games(), lfb_competition, wbbl_competition

## Examples

```
try(genius_schedule(org = "CEBL", comp_id = 47852))
```

genius_schedule_embed **Genius Schedule (Embed)**

## Description

Every match of a competition from the JSON-wrapped embed page (orgs whose plain pages answer 500).

Endpoint: GET `https://hosted.dcd.shared.geniussports.com/embednf/{org}/en/competition/{comp_id}/sched`

The same columns and rules as `genius_schedule()`, read from the `/embednf/ {css, js, html}` envelope.

## Usage

```
genius_schedule_embed(org, comp_id, round_number = -1, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| org | Genius ORG code of the league (ESAKE, CEBL, BNXT, SLB, ...); see `genius_competitions()`. |
| comp_id | Genius competition id: one per org competition-season (`comp_id` of `genius_competitions()`), e.g. 49915. |
| round_number | Round to list; -1 (default) lists every round of the competition. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): `data.json` as served plus `_request` (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json` carries no global team or player id: `tno` (1 or 2) and `pno` are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when Genius has nothing for the request (HTTP 404, the S3 `AccessDenied` 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); `wehoop_bad_request` for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); `wehoop_asset_fetch`

when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

```
genius_schedule_embed(org = "HKS", comp_id = 49946)
```

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| match_id | integer | From the block id `extfix_<id>`. Genius match id: the {matchId} of data.json and /u/; one gl |
| link_match_id | integer | Match id of the block's `/match/<id>/summary` link; equals match_id (the capture refuses a p |
| comp_id | integer | CompId of the match link, or of the team links when the org hides match pages (FFBB); equa |
| org | character | ORG code of the match link (or the team links, as comp_id) |
| status | character | Status from the block class STATUS_*: COMPLETE, SCHEDULED, POSTPONED, ... |
| date_time | character | Tip-off as displayed, 'Mon D, YYYY, h:mm AM' (local time, zone not stated); null when un |
| venue_id | integer | Genius venue id (the /venue/{id} of hosted pages); null when the venue is plain text (FFBB) |
| venue_name | character | Venue name as entered (local script) |
| home_team_id | integer | Home team: Genius team id (warehouse key space, global across orgs and seasons; the /team |
| home_team_name | character | Home team full name |
| home_team_code | character | Home team code (local script possible) |
| home_score | integer | Home points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |
| away_team_id | integer | Away team: Genius team id (warehouse key space, global across orgs and seasons; the /teamA |
| away_team_name | character | Away team full name |
| away_team_code | character | Away team code (local script possible) |
| away_score | integer | Away points; null before tip-off (typed integer only on a route whose 3 sampled rows were al |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

---

`genius_shots` **Genius Shot Chart**

---

## Description

Shot chart of a finished game: court x/y (percent of the full court), result and shooter.

Endpoint: `GET https://fibalivestats.dcd.shared.geniussports.com/data/{match_id}/data.json`

`x` and `y` are percentages (0-100) of the full court length and width, NOT normalised to one basket (a team attacks one end in periods 1-2 and the other in 3-4); the origin corner is unverified. `result` is 1 for a made shot, 0 for a miss; `action_number` joins the play-by-play row.

## Usage

```
genius_shots(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| match_id | Genius match id (match_id of genius_schedule(), genius_widget_games() or genius_competition_games()), e.g. 2902690. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the parsed body or page record (a list): data.json as served plus _request (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set options(wehoop.proxy = ...) or the GENIUS_PROXY environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (comp_id): genius_competitions() lists an org's competitions with the sdv END-year season (a START-year org lists 2025-26 as season 2026), genius_match_competition() maps a game to its competition, and data.json itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of /u/ (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. data.json carries no global team or player id: tno (1 or 2) and pno are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting wehoop_error): wehoop_no_data when Genius has nothing for the request (HTTP 404, the S3 AccessDenied 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); wehoop_bad_request for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); wehoop_asset_fetch when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (transient = TRUE: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

```
genius_shots(match_id = 2902690)
```

## Value

A wehoop_data tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when return_parsed = FALSE:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for, stamped b |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedule and widg |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In-game only: |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted); a shot join |

| | | |
|---|---|---|
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| action_type | character | Shot value: 2pt or 3pt |
| sub_type | character | Shot type (layup, drivinglayup, jumpshot, pullupjumpshot, dunk, ...) |
| result | integer | Shot result: 1 = made, 0 = missed. |
| x | double | Court x, PERCENT of the full court length (0-100, float), along the long axis. Not normalised to |
| y | double | Court y, PERCENT of the full court width (0-100, float), across the court (observed 1.7-97.2). C |
| player | character | Shooter's display name as the org enters it, in the local script when it uses one. |
| shirt_number | character | Shooter's shirt number (string) |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `lfb_competition`, `wbbl_competition`

---

`genius_team_box` **Genius Team Box Score**

---

## Description

Team box score of a finished game: score by period, fouls, timeouts and the team totals.

Endpoint: `GET https://fibalivestats.dcd.shared.geniussports.com/data/{match_id}/data.json`

One row per team (`tno` 1 and 2; the feed has no team id). Percentages are numeric 0-100. `ot_score` is the points in overtime (NA without overtime).

## Usage

```
genius_team_box(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `match_id` | Genius match id (`match_id` of `genius_schedule()`, `genius_widget_games()` or `genius_competition_games()`), e.g. 2902690. |
| `return_parsed` | Parse into a tibble (default `TRUE`); `FALSE` returns the parsed body or page record (a list): `data.json` as served plus `_request` (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the `GENIUS_PROXY` environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league

sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json` carries no global team or player id: `tno` (1 or 2) and `pno` are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when Genius has nothing for the request (HTTP 404, the S3 `AccessDenied` 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); `wehoop_bad_request` for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); `wehoop_asset_fetch` when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

```
genius_team_box(match_id = 2902690)
```

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedul |
| name | character | Team name as the org enters it |
| name_international | character | Team name in Latin script |
| short_name | character | Short team name (shortName in the feed), in the competition's own script when the c |
| code | character | Team code, three letters as entered (may be in the local script, e.g. Greek 'AEK' in C |
| code_international | character | Team code in Latin script |
| coach | character | Head coach display name |
| score | integer | Team points (current or final); a numeric string in the 2014-15 era sample (WBBL 12 |
| full_score | integer | Team points including overtime; equal to score in every sample (a numeric string in t |
| p1_score | integer | Points scored in period 1 (a numeric string in the 2014-15 era) |
| p2_score | integer | Points scored in period 2 (a numeric string in the 2014-15 era) |
| p3_score | integer | Points scored in period 3 (a numeric string in the 2014-15 era) |
| p4_score | integer | Points scored in period 4 (a numeric string in the 2014-15 era) |
| ot_score | integer | Points the team scored in overtime; null without overtime. Whether several overtimes |
| fouls | integer | Team fouls in the current (last) period |
| timeouts | integer | Timeouts counter of the current period (unverified: remaining vs taken); 0 in every sa |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '22 |
| field_goals_made | integer | Team total: field goals made |
| field_goals_attempted | integer | Team total: field goals attempted |
| field_goals_percentage | double | Team total: field-goal percentage, 0-100, rounded to an integer by the feed |

| | | |
|---|---|---|
| three_pointers_made | integer | Team total: three-pointers made |
| three_pointers_attempted | integer | Team total: three-pointers attempted |
| three_pointers_percentage | double | Team total: three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Team total: two-pointers made |
| two_pointers_attempted | integer | Team total: two-pointers attempted |
| two_pointers_percentage | double | Team total: two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Team total: free throws made |
| free_throws_attempted | integer | Team total: free throws attempted |
| free_throws_percentage | double | Team total: free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Team total: defensive rebounds |
| rebounds_offensive | integer | Team total: offensive rebounds |
| rebounds_total | integer | Team total: total rebounds |
| assists | integer | Team total: assists |
| turnovers | integer | Team total: turnovers |
| steals | integer | Team total: steals |
| blocks | integer | Team total: blocks made |
| blocks_received | integer | Team total: own shots blocked by an opponent |
| fouls_personal | integer | Team total: personal fouls committed |
| fouls_on | integer | Team total: fouls drawn |
| points | integer | Team total: points |
| points_from_turnovers | integer | Team total: points off turnovers |
| points_second_chance | integer | Team total: second-chance points |
| points_fast_break | integer | Team total: fast-break points |
| points_in_the_paint | integer | Team total: points in the paint |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no l |
| bench_points | integer | Team total: points scored by non-starters |
| biggest_lead | integer | Team total: biggest lead, points |
| biggest_scoring_run | integer | Team total: biggest unanswered scoring run, points |
| lead_changes | integer | Team total: lead changes in the game (same value on both teams) |
| times_scores_level | integer | Team total: times the score was tied (same value on both teams) |
| time_leading | double | Team total: minutes the team led, decimal minutes (33.58 = 33 min 35 s) |
| fouls_team | integer | Team total: team (bench) fouls |
| fouls_total | integer | Team total: all fouls committed (personal + technical + team bench) |
| rebounds_team | integer | Team total: team rebounds (credited to no player) |
| rebounds_team_defensive | integer | Team total: team defensive rebounds |
| rebounds_team_offensive | integer | Team total: team offensive rebounds |
| turnovers_team | integer | Team total: team turnovers (credited to no player, e.g. shot clock) |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`,

genius_schedule(), genius_schedule_embed(), genius_shots(), genius_team_crosswalk(), genius_widget_games(), lfb_competition, wbbl_competition

---

genius_team_crosswalk **Genius Team Crosswalk**

---

## Description

Sportality platform team uuid <-> Genius team id, from a Sportality frame that carries both.

The Genius ids are the ones the genius schedule frames call home_team_id / away_team_id. No request is made.

## Usage

```
genius_team_crosswalk(df, source)
```

## Arguments

| | |
|---|---|
| df | A Sportality data frame. |
| source | One of "sportality_standings", "sportality_teams", "sportality_players". |

## Details

Sources: sportality_standings (team_uuid / team_id), sportality_teams (info_uuid / info_team_id) and sportality_players (info_team_uuid / info_team_id). A wrong source, a missing column, a wrong id type or a key that maps to two ids raises wehoop_bad_request.

## Value

A wehoop_data tibble, one row per team:

| col_name | type | description |
|---|---|---|
| team_uuid | character | Sportality platform team uuid. |
| genius_team_id | integer | Genius team id of the same team (home_team_id / away_team_id of the genius schedule frames |

## See Also

Other Genius Functions: genius_competition_games(), genius_competitions(), genius_game(), genius_game_crosswalk(), genius_match_competition(), genius_pbp(), genius_player_box(), genius_schedule(), genius_schedule_embed(), genius_shots(), genius_team_box(), genius_widget_games(), lfb_competition, wbbl_competition

## Examples

```
standings <- data.frame(team_uuid = c("a1", "b2"), team_id = c(101L, 102L))
genius_team_crosswalk(standings, "sportality_standings")
```

`genius_widget_games` **Genius Widget Games**

## Description

The latest games a league site's results widget shows (needs no proxy).

Endpoint: `GET https://widget.wh.geniussports.com/widget/?{widget_key}`

About 5 to 50 games, newest competition first. `webcast_org` is the LiveStats ORG of the webcast link (`NZN` for the hosted `NZNBL`); a known widget key must return its own ORG's games.

## Usage

```
genius_widget_games(widget_key, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `widget_key` | The league site's public widget key (the bare query string of `widget.wh.geniussports.com/widget/`), e.g. `"RFJIN037I07QV520B7WEXKV1GYNA31"` (ESAKE). |
| `return_parsed` | Parse into a tibble (default `TRUE`); `FALSE` returns the parsed body or page record (a list): `data.json` as served plus `_request` (the ids asked for), or the sdv-internal-refs capture shape of a page. |

## Details

**Access.** The Genius hosts publish no robots.txt and no terms. CloudFront refuses datacenter IPs on the LiveStats and hosted hosts: from a server set `options(wehoop.proxy = ...)` or the `GENIUS_PROXY` environment variable to an http(s) proxy URL (never printed or logged). Requests are spaced at least one second apart per host; keep crawls slow and incremental. Some league sites (the NZ NBL) forbid AI crawlers in their own robots.txt, which does not cover these hosts. Unofficial use of a public source: payloads are not redistributed as release assets.

**Org and season.** Every competition is an ORG code plus a competition id (`comp_id`): `genius_competitions()` lists an org's competitions with the sdv END-year `season` (a START-year org lists 2025-26 as season 2026), `genius_match_competition()` maps a game to its competition, and `data.json` itself names no date, competition or season. The hosted ORG can differ from the LiveStats ORG of `/u/` (NZNBL vs NZN). Match ids are one global sequence across every Genius customer. `data.json` carries no global team or player id: `tno` (1 or 2) and `pno` are in-game numbers; only finished games are read.

**Errors** (classed conditions, all inheriting `wehoop_error`): `wehoop_no_data` when Genius has nothing for the request (HTTP 404, the S3 `AccessDenied` 403, an empty 200 from the hosted host, an empty recent-games window, a schedule or widget with no matches); `wehoop_bad_request` for HTTP 400 / 422 or a URL that is not a Genius route (e.g. a non-numeric id); `wehoop_asset_fetch` when the answer is unknown: the CloudFront "Request blocked" 403, a rate limit or 5xx, the intermittent HTTP 500 "A website error has occurred" page (`transient = TRUE`: retry later), a dead connection, an error envelope, a cut-off page, an unfinished game, or a body for another match, competition or ORG.

```
genius_widget_games(widget_key = "RFJIN037I07QV520B7WEXKV1GYNA31")
```

## Value

A `wehoop_data` tibble (a zero-row tibble with these columns when Genius answers an empty valid page), or the raw list when `return_parsed = FALSE`:

| col_name | type | description |
|---|---|---|
| match_id | integer | From the webcast link. Genius match id: the {matchId} of data.json and /u/; one global sequ |
| webcast_org | character | ORG code of the webcast link; can differ from the hosted ORG (NZNBL games link /webcas |
| competition_name | character | Competition label, free text |
| status | character | Status text as shown ('Final', 'Live', a tip-off time) |
| date | character | Date as shown, 'DD/MM/YYYY' in samples (format follows the widget locale; unverified fo |
| home_team_id | integer | Home team from the `tid<N>` class. Genius team id (warehouse key space, global across orgs |
| home_team_code | character | Home team short code from the widget, in the local script when the org enters one. |
| home_score | integer | Home points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |
| away_team_id | integer | Away team from the `tid<N>` class. Genius team id (warehouse key space, global across orgs |
| away_team_code | character | Away team short code from the widget, in the local script when the org enters one. |
| away_score | integer | Away points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `lfb_competition`, `wbbl_competition`

---

`helper_espn_wbb_pbp` **Parse ESPN WBB PBP, helper function**

---

## Description

**Parse ESPN WBB PBP, helper function**

**Parse ESPN WBB Team Box, helper function**

**Parse ESPN WBB Player Box, helper function**

## Usage

```
helper_espn_wbb_pbp(resp)

helper_espn_wbb_team_box(resp)

helper_espn_wbb_player_box(resp)
```

## Arguments

| | |
|---|---|
| `resp` | Response object from the ESPN WBB game summary endpoint |

## Value

For `helper_espn_wbb_pbp()`: a tibble with the columns of espn_basketball_game_all_plays_plays_schema, or `NULL` when the game has no play-by-play. `home_score` / `away_score` are rebuilt from the scoring plays when that reproduces the header's official final, otherwise they are ESPN's per-play values; the last column, `score_source` (`"rebuilt"` / `"espn"`), says which.

Returns a tibble

Returns a tibble

---

`helper_espn_wnba_pbp` **Parse ESPN WNBA PBP, helper function**

---

## Description

**Parse ESPN WNBA PBP, helper function**

**Parse ESPN WNBA Team Box, helper function**

**Parse ESPN WNBA Player Box, helper function**

## Usage

```
helper_espn_wnba_pbp(resp)

helper_espn_wnba_team_box(resp)

helper_espn_wnba_player_box(resp)
```

## Arguments

| | |
|---|---|
| `resp` | Response object from the ESPN WNBA game summary endpoint |

## Value

For `helper_espn_wnba_pbp()`: a tibble with the columns of espn_basketball_game_all_plays_plays_schema, or `NULL` when the game has no play-by-play. `home_score` / `away_score` are rebuilt from the scoring plays when that reproduces the header's official final, otherwise they are ESPN's per-play values; the last column, `score_source` (`"rebuilt"` / `"espn"`), says which.

Returns a tibble

Returns a tibble

`hhs_team_roster` **Her Hoop Stats Team Roster**

## Description

**Get a team's player roster + stats from Her Hoop Stats.**

Requires a Her Hoop Stats subscription (see `hhs_teams()`).

## Usage

```
hhs_team_roster(team_link, email = NULL, password = NULL)
```

## Arguments

| | |
|---|---|
| `team_link` | A team page path or URL (the `team_link` column from `hhs_teams()`). |
| `email`, `password` | Optional explicit subscription credentials. |

## Value

A `wehoop_data` tibble, one row per player.

## Note

Requires a Her Hoop Stats subscription. Member-table layout may change.

## See Also

Other Her Hoop Stats Functions: `hhs_team_stats()`, `hhs_teams()`

## Examples

```
## Not run:
  teams <- hhs_teams(2024, 2024, 1)
  hhs_team_roster(team_link = teams$team_link[1])

## End(Not run)
```

hhs_team_stats **Her Hoop Stats Team Statistics**

## Description

**Get a team's statistics page from Her Hoop Stats.**

Requires a Her Hoop Stats subscription (see hhs_teams()).

## Usage

```
hhs_team_stats(team_link, email = NULL, password = NULL)
```

## Arguments

| | |
|---|---|
| team_link | A team page path or URL, e.g. the team_link column returned by hhs_teams(). |
| email, password | Optional explicit subscription credentials. |

## Value

A wehoop_data tibble of the team's primary statistics table.

## Note

Requires a Her Hoop Stats subscription. Member-table layout may change.

## See Also

Other Her Hoop Stats Functions: hhs_team_roster(), hhs_teams()

## Examples

```
## Not run:
  teams <- hhs_teams(2024, 2024, 1)
  hhs_team_stats(team_link = teams$team_link[1])

## End(Not run)
```

hhs_teams **Her Hoop Stats NCAA Team Single-Season Index**

## Description

**Get the NCAA women's team single-season summary table from Her Hoop Stats.**

Her Hoop Stats is a *subscription* service; this logs in with your own credentials (read from the email/password arguments, the wehoop.herhoopstats_email/..._password options, or the HERHOOPSTATS_EMAIL/HERHO environment variables) and returns one row per team-season, with a team_link column for hhs_team_stats() / hhs_team_roster().

## Usage

```
hhs_teams(
  min_season = most_recent_wbb_season(),
  max_season = most_recent_wbb_season(),
  division = 1,
  email = NULL,
  password = NULL
)
```

## Arguments

| | |
|---|---|
| min_season, max_season | Season range (4-digit ending year; 2024 = 2023-24). Default both to most_recent_wbb_season(). |
| division | NCAA division: 1 (default), 2 or 3. |
| email, password | Optional explicit subscription credentials. |

## Value

A wehoop_data tibble, one row per team-season (record, scoring and per-100-possession columns), plus a team_link column and the requested min_season / max_season / division.

## Note

Requires a Her Hoop Stats subscription. The member-table layout may change over time; selectors here are a starting point.

## See Also

Other Her Hoop Stats Functions: hhs_team_roster(), hhs_team_stats()

## Examples

```
## Not run:
  # requires HERHOOPSTATS_EMAIL / HERHOOPSTATS_PW
  hhs_teams(min_season = 2024, max_season = 2024, division = 1)

## End(Not run)
```

---

`lfb_competition` **Ligue Feminine de Basket (France, FFBB) (Genius LiveStats)**

---

## Description

Convenience wrappers binding the Genius family to the Ligue Feminine de Basket (France, FFBB) (women's; Genius ORG `FFBB`, LiveStats ORG `FFBB`). Season rule: END year; Genius year + 1 (2018-19 = 2019); Genius hosts only 2018 and 2019.

- `lfb_competitions()`: every competition of the `FFBB` org, with its END-year season (`genius_competitions()`).
- `lfb_schedule()`: every match of one season (`genius_schedule()`).
- `lfb_team_box()`, `lfb_player_box()`, `lfb_pbp()`, `lfb_shots()`: one finished game (`genius_team_box()`, `genius_player_box()`, `genius_pbp()`, `genius_shots()`).

## Usage

```
lfb_competitions(return_parsed = TRUE)

lfb_schedule(season = NULL, comp_id = NULL, round_number = -1, return_parsed = TRUE)

lfb_team_box(match_id, return_parsed = TRUE)

lfb_player_box(match_id, return_parsed = TRUE)

lfb_pbp(match_id, return_parsed = TRUE)

lfb_shots(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| `season` | END-year season (see the season rule). `NULL` (default) = the newest season the org lists. |
| `comp_id` | Genius competition id; skips the season lookup (needed when the label rule matches several competitions). |
| `round_number` | Round to list; -1 (default) lists every round. |
| `match_id` | Genius match id (`match_id` of `lfb_schedule()`) of a game of this competition. |
| `return_parsed` | Parse into a tibble (default `TRUE`); `FALSE` returns the raw list. |

## Details

The season's competition is picked from the FFBB chooser by the label rule include = `^ligue feminine( de basket)?$`, exclude = `^$` (case-insensitive); cups and play-offs have their own `comp_id`. Several matching competitions raise `wehoop_bad_request` (pass `comp_id`); none raises `wehoop_no_data`.

Genius match ids are one global sequence shared by every customer and `data.json` names no competition, so each game alias checks the id on the game's LiveStats shell (`/u/FFBB/<match_id>/`: one extra request per game, cached per process for an hour): its competition name must pass the label rule. Another org's game, a cup or play-off game of FFBB, or a shell request that fails twice (or names no competition) raises `wehoop_asset_fetch`. The base `genius_*()` function reads any game unchecked.

Access, proxy and error notes: see `genius_game()`.

```
lfb_competitions()
lfb_schedule()
```

## Value

The base function's `wehoop_data` tibble, same columns and types (the raw list when `return_parsed = FALSE`), one table per function:

`lfb_competitions()`:

The columns of `genius_competitions()`:

| col_name | type | description |
|---|---|---|
| org | character | ORG code the request used (the hosted ORG, or the LiveStats ORG for the /u/ shell). |
| comp_id | integer | Genius competition id (compId): one per org competition-season; the {compId} of hosted and data/ |
| label | character | Competition label as the org typed it, free text in any language: may or may not carry a season ('BA |
| league_id | integer | Genius league id of the org (ESAKE = 115; IBBA has two: 187 and 549) |
| year | integer | Genius season year of the competition: the START year of a two-year season (BNXT '2023-2024' = |
| season | integer | sdv END-year season of the competition (2026 = 2025-26) from the label span, else the Genius yea |
| season_rule | character | Which rule produced season: 'label span', 'genius year + 1 (split)', 'genius year (calendar)', 'label |
| selected | logical | True for the competition the hosted chooser pre-selects (the org's default, not always the newest). |

`lfb_schedule()`:

The columns of `genius_schedule()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | From the block id `extfix_<id>`. Genius match id: the {matchId} of data.json and /u/; one gl |
| link_match_id | integer | Match id of the block's `/match/<id>/summary` link; equals match_id (the capture refuses a p |
| comp_id | integer | CompId of the match link, or of the team links when the org hides match pages (FFBB); equa |
| org | character | ORG code of the match link (or the team links, as comp_id) |
| status | character | Status from the block class STATUS_*: COMPLETE, SCHEDULED, POSTPONED, ... |
| date_time | character | Tip-off as displayed, 'Mon D, YYYY, h:mm AM' (local time, zone not stated); null when un |
| venue_id | integer | Genius venue id (the /venue/{id} of hosted pages); null when the venue is plain text (FFBB) |
| venue_name | character | Venue name as entered (local script) |
| home_team_id | integer | Home team: Genius team id (warehouse key space, global across orgs and seasons; the /team |

| | | |
|---|---|---|
| home_team_name | character | Home team full name |
| home_team_code | character | Home team code (local script possible) |
| home_score | integer | Home points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |
| away_team_id | integer | Away team: Genius team id (warehouse key space, global across orgs and seasons; the /team/ |
| away_team_name | character | Away team full name |
| away_team_code | character | Away team code (local script possible) |
| away_score | integer | Away points; null before tip-off (typed integer only on a route whose 3 sampled rows were al |

`lfb_team_box()`:

The columns of `genius_team_box()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedul |
| name | character | Team name as the org enters it |
| name_international | character | Team name in Latin script |
| short_name | character | Short team name (shortName in the feed), in the competition's own script when the c |
| code | character | Team code, three letters as entered (may be in the local script, e.g. Greek 'AEK' in C |
| code_international | character | Team code in Latin script |
| coach | character | Head coach display name |
| score | integer | Team points (current or final); a numeric string in the 2014-15 era sample (WBBL 12 |
| full_score | integer | Team points including overtime; equal to score in every sample (a numeric string in t |
| p1_score | integer | Points scored in period 1 (a numeric string in the 2014-15 era) |
| p2_score | integer | Points scored in period 2 (a numeric string in the 2014-15 era) |
| p3_score | integer | Points scored in period 3 (a numeric string in the 2014-15 era) |
| p4_score | integer | Points scored in period 4 (a numeric string in the 2014-15 era) |
| ot_score | integer | Points the team scored in overtime; null without overtime. Whether several overtimes |
| fouls | integer | Team fouls in the current (last) period |
| timeouts | integer | Timeouts counter of the current period (unverified: remaining vs taken); 0 in every sa |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '22 |
| field_goals_made | integer | Team total: field goals made |
| field_goals_attempted | integer | Team total: field goals attempted |
| field_goals_percentage | double | Team total: field-goal percentage, 0-100, rounded to an integer by the feed |
| three_pointers_made | integer | Team total: three-pointers made |
| three_pointers_attempted | integer | Team total: three-pointers attempted |
| three_pointers_percentage | double | Team total: three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Team total: two-pointers made |
| two_pointers_attempted | integer | Team total: two-pointers attempted |
| two_pointers_percentage | double | Team total: two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Team total: free throws made |
| free_throws_attempted | integer | Team total: free throws attempted |
| free_throws_percentage | double | Team total: free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Team total: defensive rebounds |
| rebounds_offensive | integer | Team total: offensive rebounds |
| rebounds_total | integer | Team total: total rebounds |
| assists | integer | Team total: assists |

| | | |
|---|---|---|
| turnovers | integer | Team total: turnovers |
| steals | integer | Team total: steals |
| blocks | integer | Team total: blocks made |
| blocks_received | integer | Team total: own shots blocked by an opponent |
| fouls_personal | integer | Team total: personal fouls committed |
| fouls_on | integer | Team total: fouls drawn |
| points | integer | Team total: points |
| points_from_turnovers | integer | Team total: points off turnovers |
| points_second_chance | integer | Team total: second-chance points |
| points_fast_break | integer | Team total: fast-break points |
| points_in_the_paint | integer | Team total: points in the paint |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no l |
| bench_points | integer | Team total: points scored by non-starters |
| biggest_lead | integer | Team total: biggest lead, points |
| biggest_scoring_run | integer | Team total: biggest unanswered scoring run, points |
| lead_changes | integer | Team total: lead changes in the game (same value on both teams) |
| times_scores_level | integer | Team total: times the score was tied (same value on both teams) |
| time_leading | double | Team total: minutes the team led, decimal minutes (33.58 = 33 min 35 s) |
| fouls_team | integer | Team total: team (bench) fouls |
| fouls_total | integer | Team total: all fouls committed (personal + technical + team bench) |
| rebounds_team | integer | Team total: team rebounds (credited to no player) |
| rebounds_team_defensive | integer | Team total: team defensive rebounds |
| rebounds_team_offensive | integer | Team total: team offensive rebounds |
| turnovers_team | integer | Team total: team turnovers (credited to no player, e.g. shot clock) |

`lfb_player_box()`:

The columns of `genius_player_box()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked fo |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |
| name | character | Display name ('T. Nichols') |
| first_name | character | Given name in the competition's own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition's own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name ('T. Nichols'), in the local script when the org enters one |
| shirt_number | character | Shirt number as a string ('0' and '00' differ) |
| playing_position | character | Playing position code (PG, SG, SF, PF, C, G, F, ...); empty string when not entered |
| starter | integer | 1 when the player started the game, else 0 |

| | | |
|---|---|---|
| captain | integer | 1 for the team captain, else 0; absent in some orgs |
| active | integer | 1 when the player is on court now (at game end: on court at the final whistle), else 0 |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '2 |
| field_goals_made | integer | Field goals the player made in the game (two- and three-pointers together; a count). |
| field_goals_attempted | integer | Field goals the player attempted in the game (two- and three-pointers together; a cou |
| field_goals_percentage | double | Field-goal percentage, 0-100, rounded to an integer by the feed |
| three_pointers_made | integer | Three-pointers made |
| three_pointers_attempted | integer | Three-pointers attempted |
| three_pointers_percentage | double | Three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Two-pointers made |
| two_pointers_attempted | integer | Two-pointers attempted |
| two_pointers_percentage | double | Two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Free throws the player made in the game (a count). |
| free_throws_attempted | integer | Free throws the player attempted in the game (a count). |
| free_throws_percentage | double | Free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Defensive rebounds |
| rebounds_offensive | integer | Offensive rebounds |
| rebounds_total | integer | Total rebounds (offensive plus defensive) the player collected in the game. |
| assists | integer | Assists credited to the player in the game. |
| turnovers | integer | Turnovers committed by the player in the game. |
| steals | integer | Steals credited to the player in the game. |
| blocks | integer | Shots the player blocked in the game (blocks made, not received). |
| blocks_received | integer | Own shots blocked by an opponent |
| fouls_personal | integer | Personal fouls committed |
| fouls_on | integer | Fouls drawn: personal fouls committed against the player in the game. |
| points | integer | Points the player scored in the game, free throws included. |
| points_from_turnovers | integer | Points off turnovers |
| points_second_chance | integer | Second-chance points |
| points_fast_break | integer | Fast-break points |
| points_in_the_paint | integer | Points the player scored in the paint in the game (the feed's sPointsInThePaint). |
| plus_minus_points | integer | Plus/minus: team point margin while the player was on court |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no |

`lfb_pbp()`:

The columns of `genius_pbp()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| row_index | integer | Position of the action in the feed order, 0 = the newest action by game time (normall |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted) |

| | | |
|---|---|---|
| previous_action | integer | action_number of the linked earlier action (the shot an assist or rebound belongs to); |
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| gt | character | Game clock, 'mm:ss' remaining in the period |
| clock | character | Game clock with hundredths, 'mm:ss:cc' remaining in the period; absent in older fe |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |
| player | character | Player display name; '' when no player |
| shirt_number | character | Player shirt number (string); '' when no player |
| first_name | character | Given name in the competition's own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition's own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name ('T. Nichols'), in the local script when the org enters one |
| action_type | character | Action type: 2pt, 3pt, freethrow, rebound, assist, turnover, steal, block, foul, foulon ( |
| sub_type | character | Action subtype, free-text code per actionType: shot type for 2pt/3pt (layup, drivingla |
| qualifier | character | Qualifier codes of the action joined with commas (pointsinthepaint, fromturnover, 2r |
| success | integer | 1 = made / successful, 0 = missed (shots, free throws); 1 for other actions |
| scoring | integer | 1 when the action is a scoring attempt (shot or free throw), else 0 |
| score_1 | integer | Team 1 score after the action, always an integer (the feed sends a string in most gam |
| score_2 | integer | Team 2 score after the action, always an integer (the feed sends a string in most gam |
| lead | integer | Score margin after the action, team 1 minus team 2 (negative when team 2 leads) |

`lfb_shots()`:
The columns of `genius_shots()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for, stamped b |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedule and widg |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In-game only: |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted); a shot join |
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| action_type | character | Shot value: 2pt or 3pt |
| sub_type | character | Shot type (layup, drivinglayup, jumpshot, pullupjumpshot, dunk, ...) |
| result | integer | Shot result: 1 = made, 0 = missed. |
| x | double | Court x, PERCENT of the full court length (0-100, float), along the long axis. Not normalised to |
| y | double | Court y, PERCENT of the full court width (0-100, float), across the court (observed 1.7-97.2). C |
| player | character | Shooter's display name as the org enters it, in the local script when it uses one. |
| shirt_number | character | Shooter's shirt number (string) |

## See Also

Other Genius Functions: `genius_competition_games()`, `genius_competitions()`, `genius_game()`, `genius_game_crosswalk()`, `genius_match_competition()`, `genius_pbp()`, `genius_player_box()`, `genius_schedule()`, `genius_schedule_embed()`, `genius_shots()`, `genius_team_box()`, `genius_team_crosswalk()`, `genius_widget_games()`, `wbbl_competition`

lmb_competition **Liga Mayor de Baloncesto (El Salvador) (Atrium shortcuts)**

## Description

**Per-competition shortcuts over the Atrium embed API for Liga Mayor de Baloncesto (El Salvador) (website_id = 13; men's and women's).**

Thin bindings of the parameterised `atrium_*()` wrappers to this site, minted from the competition registry (sdv-py `tools/codegen/basketball_competitions.yaml`); the full surface stays on the `atrium_*()` wrappers.

## Usage

```
lmb_seasons(locale = NULL)
lmb_schedule(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lmb_standings(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lmb_teams(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lmb_player_box(fixture_id, locale = NULL)
lmb_team_box(fixture_id, locale = NULL)
lmb_pbp(fixture_id, locale = NULL)
lmb_shots(fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `season` | (*integer* optional): Season END year (season rule below). Omit `season` and `season_id` for the site's current season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in the seasons listing); skips the END-year lookup. |
| `competition_id` | (*character* optional): Competition UUID (a site carries many: conferences, cups); needed when several share the season. |
| `locale` | (*character* optional): Label locale, e.g. `"en-EN"`; omitted = the site's own language. |
| `fixture_id` | (*character* required): Fixture UUID (`fixture_id` in the schedule). |

## Details

Season rule: calendar year.

Errors and terms of use are those of `atrium_fixtures()`: unofficial use of the public widget endpoint, at least 1 s between requests, not republished as release assets. The site also serves women's competitions (pass their `competition_id`).

```
lmb_seasons()
lmb_schedule()
```

## Value

A `wehoop_data` tibble with the columns of the base wrapper, one table per function:

`lmb_seasons()`:

The columns of `atrium_seasons()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, |
| season_id | character | Season UUID; pass it as season_id to any season-scoped wrapper. |
| competition_id | character | Competition UUID; pass it as competition_id to pick one of several seasons of an END year. |
| competition_name | character | Competition name in the site's language; the same name can appear for several years. |
| season_name | character | Season name in the site's language, e.g. Betclic ELITE 2025. |
| abbreviation_local | character | Local abbreviation of the season; null in every body seen. |
| year | integer | The site's own year of the season: the START year on LNB (site 12), the calendar year elsew |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year on th |
| is_default_season | logical | True for the season the body is about (the site's default when none was asked). |

`lmb_schedule()`:

The columns of `atrium_fixtures()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| season_id | character | Season UUID the page is about (the site's default season when none was asked). |
| competition_id | character | Competition UUID of the season. |
| competition_name | character | Competition name in the site's language, e.g. Betclic ELITE or LIGA CALIENTE.M |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar yea |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| name | character | Fixture name as the site prints it (e.g. Paris - Lyon-Villeurbanne); null on LNBP. |
| round | character | Round of the game as the site words it, e.g. Journee: 30. |
| round_code | character | Round code; null in every body seen. |
| pool | character | Pool the game belongs to; null when the competition has none. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a s |
| stage_code | character | Opaque 30-hex stage code of the game; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (`YYYY-MM-DDTHH:MM:SS`, |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_loc |
| times_unconfirmed | logical | True while the start time is not final. |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |

| | | |
|---|---|---|
| home_result_place | integer | Result place of the home team: 1 for the winner, 2 for the loser. |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_draw | logical | True when the game ended level for the home team. |
| home_shootout_attempts | character | Shootout attempts of the home team as sent (an empty string becomes null). |
| home_secondary_score | character | Secondary score of the home team (unused in the samples; shown as text). |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_place | integer | Result place of the away team: 1 for the winner, 2 for the loser. |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_draw | logical | True when the game ended level for the away team. |
| away_shootout_attempts | character | Shootout attempts of the away team as sent (an empty string becomes null). |
| away_secondary_score | character | Secondary score of the away team (unused in the samples; shown as text). |

`lmb_standings()`:

The columns of `atrium_standings()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year c |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and l |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

`lmb_teams()`:

The columns of `atrium_teams()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Au |
| season_id | character | Season UUID of the team list. |
| entity_id | character | Team UUID (entity): one id per club across seasons, shared by every site that shows the team. |
| name | character | Team name in the site's language. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

`lmb_player_box()`:

The columns of `atrium_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID the player is rostered with for this game. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| person_name | character | Player name as the site prints it (upper case on LNBP and NBL1). |
| bib | character | Jersey number the player wears, as text. |
| position | character | Position code on the roster (LNB, e.g. PG); null on LNBP. |
| starter | logical | True when the player started the game. |
| active | logical | True when the player is on the court in a live game. |
| participated | logical | True when the player entered the game; false rows have null statistics. |
| did_not_play_reason | character | Reason the player did not play; null in every body seen. |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, this game). |
| points | integer | Points scored (player, this game). |
| rebounds | integer | Total rebounds (player, this game). |
| rebounds_offensive | integer | Offensive rebounds (player, this game). |
| rebounds_defensive | integer | Defensive rebounds (player, this game). |
| assists | integer | Assists (player, this game). |
| steals | integer | Steals (player, this game). |
| blocks | integer | Shots blocked (player, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, this game). |
| turnovers | integer | Turnovers (player, this game). |
| fouls_total | integer | Fouls committed (player, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, this game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) ( |
| efficiency | integer | Efficiency rating, from the site's own formula (player, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, this game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, this game). |

| | | |
|---|---|---|
| points_two_made | integer | Two-point field goals made (player, this game). |
| points_two_attempted | integer | Two-point field goals attempted (player, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, this game). |
| points_three_made | integer | Three-point field goals made (player, this game). |
| points_three_attempted | integer | Three-point field goals attempted (player, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, this game). |
| free_throws_made | integer | Free throws made (player, this game). |
| free_throws_attempted | integer | Free throws attempted (player, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, this game). |

`lmb_team_box()`:

The columns of `atrium_team_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 2 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail route |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID (from the fixture's isHome flag). |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome fla |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its pla |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (team, this game). |
| points | integer | Points scored (team, this game). |
| rebounds | integer | Total rebounds (team, this game). |
| rebounds_offensive | integer | Offensive rebounds (team, this game). |
| rebounds_defensive | integer | Defensive rebounds (team, this game). |
| assists | integer | Assists (team, this game). |
| steals | integer | Steals (team, this game). |
| blocks | integer | Shots blocked (team, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (team, this game). |
| turnovers | integer | Turnovers (team, this game). |
| fouls_total | integer | Fouls committed (team, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (team, this game |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lin |
| efficiency | integer | Efficiency rating, from the site's own formula (team, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (team, this game |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (team, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, this game). |
| points_two_made | integer | Two-point field goals made (team, this game). |
| points_two_attempted | integer | Two-point field goals attempted (team, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (team, this game). |
| points_three_made | integer | Three-point field goals made (team, this game). |
| points_three_attempted | integer | Three-point field goals attempted (team, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (team, this game). |
| free_throws_made | integer | Free throws made (team, this game). |
| free_throws_attempted | integer | Free throws attempted (team, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (team, this game). |

| | | |
|---|---|---|
| points_fast_break | integer | Fast-break points (LNBP only) (team, this game). |
| extra_fouls_total | integer | Fouls committed (team, credited to the team rather than a player). |
| extra_rebounds | integer | Total rebounds (team, credited to the team rather than a player). |
| extra_rebounds_defensive | integer | Defensive rebounds (team, credited to the team rather than a player). |
| extra_rebounds_offensive | integer | Offensive rebounds (team, credited to the team rather than a player). |
| extra_turnovers | integer | Turnovers (team, credited to the team rather than a player). |
| total_biggest_lead | integer | Biggest lead of the game, in points (team, game total). |
| total_biggest_scoring_run | integer | Longest unanswered scoring run of the game, in points (team, game total). |
| total_field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, game total). |
| total_free_throws_percentage | double | Free-throw percentage, 0-100 (team, game total). |
| total_lead_changes | integer | Number of lead changes in the game (team, game total). |
| total_points_fast_break | integer | Fast-break points (LNBP only) (team, game total). |
| total_points_from_bench | integer | Points scored by non-starters (LNBP only) (team, game total). |
| total_points_from_turnover | integer | Points scored off opponent turnovers (LNBP only) (team, game total). |
| total_time_in_lead | character | Time the team led, as an ISO 8601 duration (PT36M11S) (team, game total). |
| total_time_in_lead_seconds | integer | Time the team led, in whole seconds (team, game total). |
| total_turnovers | integer | Turnovers (team, game total). |

`lmb_pbp()`:

The columns of `atrium_pbp()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the event in the game (periods in order, events chronological). |
| event_id | character | Event UUID, shared with the same event in atrium_shots. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID; null for team events (time-outs, team rebounds). |
| person_name | character | Player the event is credited to; null for team events (time-outs, team rebounds). |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when a shot or free throw was made, false when missed; null for other events. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

`lmb_shots()`:

The columns of `atrium_shots()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the shot among the shots of the game (not the pbp sequence). |
| event_id | character | Event UUID of the shot; the same event in atrium_pbp. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID of the shooter. |
| person_name | character | Name of the shooter. |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when the shot was made, false when missed. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_team_stats()`, `atrium_teams()`, `lnbp_competition`, `lpb_competition`, `nbl1_competition`

---

# lnbp_competition — LNBP (Mexico, with COPA VALUE and LNBP FEMENIL) (Atrium shortcuts)

---

## Description

**Per-competition shortcuts over the Atrium embed API for LNBP (Mexico, with COPA VALUE and LNBP FEMENIL) (`website_id = 14`; men's and women's).**

Thin bindings of the parameterised `atrium_*()` wrappers to this site, minted from the competition registry (sdv-py `tools/codegen/basketball_competitions.yaml`); the full surface stays on the `atrium_*()` wrappers.

## Usage

```
lnbp_seasons(locale = NULL)
lnbp_schedule(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lnbp_standings(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lnbp_teams(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lnbp_player_box(fixture_id, locale = NULL)
lnbp_team_box(fixture_id, locale = NULL)
lnbp_pbp(fixture_id, locale = NULL)
lnbp_shots(fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| season | (*integer* optional): Season END year (season rule below). Omit season and season_id for the site's current season. |
| season_id | (*character* optional): Season UUID (season_id in the seasons listing); skips the END-year lookup. |
| competition_id | (*character* optional): Competition UUID (a site carries many: conferences, cups); needed when several share the season. |
| locale | (*character* optional): Label locale, e.g. "en-EN"; omitted = the site's own language. |
| fixture_id | (*character* required): Fixture UUID (fixture_id in the schedule). |

## Details

Season rule: calendar year.

Errors and terms of use are those of atrium_fixtures(): unofficial use of the public widget endpoint, at least 1 s between requests, not republished as release assets. The site also serves women's competitions (pass their competition_id).

```
lnbp_seasons()
lnbp_schedule()
```

## Value

A wehoop_data tibble with the columns of the base wrapper, one table per function:

lnbp_seasons():

The columns of atrium_seasons():

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, |
| season_id | character | Season UUID; pass it as season_id to any season-scoped wrapper. |
| competition_id | character | Competition UUID; pass it as competition_id to pick one of several seasons of an END year. |
| competition_name | character | Competition name in the site's language; the same name can appear for several years. |
| season_name | character | Season name in the site's language, e.g. Betclic ELITE 2025. |
| abbreviation_local | character | Local abbreviation of the season; null in every body seen. |
| year | integer | The site's own year of the season: the START year on LNB (site 12), the calendar year elsew |

| | | |
|---|---|---|
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year on the |
| is_default_season | logical | True for the season the body is about (the site's default when none was asked). |

`lnbp_schedule()`:

The columns of `atrium_fixtures()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| season_id | character | Season UUID the page is about (the site's default season when none was asked). |
| competition_id | character | Competition UUID of the season. |
| competition_name | character | Competition name in the site's language, e.g. Betclic ELITE or LIGA CALIENTE.MX |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| name | character | Fixture name as the site prints it (e.g. Paris - Lyon-Villeurbanne); null on LNBP. |
| round | character | Round of the game as the site words it, e.g. Journee: 30. |
| round_code | character | Round code; null in every body seen. |
| pool | character | Pool the game belongs to; null when the competition has none. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a s |
| stage_code | character | Opaque 30-hex stage code of the game; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (YYYY-MM-DDTHH:MM:SS, |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_loca |
| times_unconfirmed | logical | True while the start time is not final. |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_place | integer | Result place of the home team: 1 for the winner, 2 for the loser. |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_draw | logical | True when the game ended level for the home team. |
| home_shootout_attempts | character | Shootout attempts of the home team as sent (an empty string becomes null). |
| home_secondary_score | character | Secondary score of the home team (unused in the samples; shown as text). |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_place | integer | Result place of the away team: 1 for the winner, 2 for the loser. |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_draw | logical | True when the game ended level for the away team. |
| away_shootout_attempts | character | Shootout attempts of the away team as sent (an empty string becomes null). |
| away_secondary_score | character | Secondary score of the away team (unused in the samples; shown as text). |

`lnbp_standings()`:
The columns of `atrium_standings()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year c |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and l |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

`lnbp_teams()`:
The columns of `atrium_teams()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Au |
| season_id | character | Season UUID of the team list. |
| entity_id | character | Team UUID (entity): one id per club across seasons, shared by every site that shows the team. |
| name | character | Team name in the site's language. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

`lnbp_player_box()`:
The columns of `atrium_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID the player is rostered with for this game. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| person_name | character | Player name as the site prints it (upper case on LNBP and NBL1). |
| bib | character | Jersey number the player wears, as text. |
| position | character | Position code on the roster (LNB, e.g. PG); null on LNBP. |
| starter | logical | True when the player started the game. |
| active | logical | True when the player is on the court in a live game. |
| participated | logical | True when the player entered the game; false rows have null statistics. |
| did_not_play_reason | character | Reason the player did not play; null in every body seen. |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, this game). |
| points | integer | Points scored (player, this game). |
| rebounds | integer | Total rebounds (player, this game). |
| rebounds_offensive | integer | Offensive rebounds (player, this game). |
| rebounds_defensive | integer | Defensive rebounds (player, this game). |
| assists | integer | Assists (player, this game). |
| steals | integer | Steals (player, this game). |
| blocks | integer | Shots blocked (player, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, this game). |
| turnovers | integer | Turnovers (player, this game). |
| fouls_total | integer | Fouls committed (player, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, this game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) ( |
| efficiency | integer | Efficiency rating, from the site's own formula (player, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, this game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, this game). |
| points_two_made | integer | Two-point field goals made (player, this game). |
| points_two_attempted | integer | Two-point field goals attempted (player, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, this game). |
| points_three_made | integer | Three-point field goals made (player, this game). |
| points_three_attempted | integer | Three-point field goals attempted (player, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, this game). |
| free_throws_made | integer | Free throws made (player, this game). |
| free_throws_attempted | integer | Free throws attempted (player, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, this game). |

`lnbp_team_box()`:

The columns of `atrium_team_box_score()`:

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 2 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail route |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID (from the fixture's isHome flag). |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome fla |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its pla |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (team, this game). |
| points | integer | Points scored (team, this game). |
| rebounds | integer | Total rebounds (team, this game). |
| rebounds_offensive | integer | Offensive rebounds (team, this game). |
| rebounds_defensive | integer | Defensive rebounds (team, this game). |
| assists | integer | Assists (team, this game). |
| steals | integer | Steals (team, this game). |
| blocks | integer | Shots blocked (team, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (team, this game). |
| turnovers | integer | Turnovers (team, this game). |
| fouls_total | integer | Fouls committed (team, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (team, this game |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lin |
| efficiency | integer | Efficiency rating, from the site's own formula (team, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (team, this game |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (team, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, this game). |
| points_two_made | integer | Two-point field goals made (team, this game). |
| points_two_attempted | integer | Two-point field goals attempted (team, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (team, this game). |
| points_three_made | integer | Three-point field goals made (team, this game). |
| points_three_attempted | integer | Three-point field goals attempted (team, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (team, this game). |
| free_throws_made | integer | Free throws made (team, this game). |
| free_throws_attempted | integer | Free throws attempted (team, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (team, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (team, this game). |
| extra_fouls_total | integer | Fouls committed (team, credited to the team rather than a player). |
| extra_rebounds | integer | Total rebounds (team, credited to the team rather than a player). |
| extra_rebounds_defensive | integer | Defensive rebounds (team, credited to the team rather than a player). |
| extra_rebounds_offensive | integer | Offensive rebounds (team, credited to the team rather than a player). |
| extra_turnovers | integer | Turnovers (team, credited to the team rather than a player). |
| total_biggest_lead | integer | Biggest lead of the game, in points (team, game total). |
| total_biggest_scoring_run | integer | Longest unanswered scoring run of the game, in points (team, game total). |
| total_field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, game total). |
| total_free_throws_percentage | double | Free-throw percentage, 0-100 (team, game total). |
| total_lead_changes | integer | Number of lead changes in the game (team, game total). |
| total_points_fast_break | integer | Fast-break points (LNBP only) (team, game total). |
| total_points_from_bench | integer | Points scored by non-starters (LNBP only) (team, game total). |
| total_points_from_turnover | integer | Points scored off opponent turnovers (LNBP only) (team, game total). |
| total_time_in_lead | character | Time the team led, as an ISO 8601 duration (PT36M11S) (team, game total). |
| total_time_in_lead_seconds | integer | Time the team led, in whole seconds (team, game total). |

| | | |
|---|---|---|
| total_turnovers | integer | Turnovers (team, game total). |

`lnbp_pbp()`:

The columns of `atrium_pbp()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the event in the game (periods in order, events chronological). |
| event_id | character | Event UUID, shared with the same event in atrium_shots. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID; null for team events (time-outs, team rebounds). |
| person_name | character | Player the event is credited to; null for team events (time-outs, team rebounds). |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when a shot or free throw was made, false when missed; null for other events. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

`lnbp_shots()`:

The columns of `atrium_shots()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the shot among the shots of the game (not the pbp sequence). |
| event_id | character | Event UUID of the shot; the same event in atrium_pbp. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID of the shooter. |
| person_name | character | Name of the shooter. |
| bib | character | Jersey number of the player, as text; null for team events. |

| | | |
|---|---|---|
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when the shot was made, false when missed. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_team_stats()`, `atrium_teams()`, `lmb_competition`, `lpb_competition`, `nbl1_competition`

---

`load_ncaa_wbb_pbp` **Load cleaned NCAA women's college basketball play-by-play from the data repo**

---

## Description

Loads season-level NCAA women's college basketball play-by-play. One row per event, with reconstructed 5-man lineups, possession numbering, shot context (transition/paint/second-chance), and an ESPN game-id crosswalk. Produced by the sdv-py `ncaa_wbb` engine (stats.ncaa.org play-by-play parsed, cleaned, and enriched); backed by the `ncaa-wbb-hoops-data` pipeline, published to the `ncaa_wbb_pbp` release tag as csv.gz/parquet/rds.

Loads shot-chart events for NCAA women's college basketball – one row per field goal attempt, with court coordinates, distance, shot zone, and made/missed result. Coordinates are only available from the 2019 season onward (stats.ncaa.org did not publish shot-location data before then), so `seasons` has a 2019 floor even though most sibling NCAA WBB datasets start at 2010. Produced by the sdv-py `ncaa_wbb` engine; backed by the `ncaa-wbb-hoops-data` pipeline, published to the `ncaa_wbb_shots` release tag as csv.gz/parquet/rds.

Loads reconstructed 5-man lineup stints for NCAA women's college basketball – one row per contiguous stretch a specific 5-player lineup was on court for a team, with on/off scoring, shooting, and rebounding splits plus the matching opponent (`opp_*`) splits. Produced by the sdv-py `ncaa_wbb` engine's lineup reconstructor; backed by the `ncaa-wbb-hoops-data` pipeline, published to the `ncaa_wbb_lineups` release tag as csv.gz/parquet/rds.

Loads reconstructed 10-man matchup stints (the home 5-man lineup crossed with the away 5-man lineup) for NCAA women's college basketball – one row per contiguous stretch both lineups were on court together, with the score at stint start/end and both team's lineup rosters. Produced by the

sdv-py ncaa_wbb engine's lineup reconstructor; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_matchup_stints release tag as csv.gz/parquet/rds.

Loads possession-level data for NCAA women's college basketball – one row per offensive possession, with the on-court 5-man lineups for both teams, points scored, assist/transition/garbage-time flags, and the possession's first and last event types. Produced by the sdv-py ncaa_wbb engine's possession detector; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_possessions release tag as csv.gz/parquet/rds.

Loads within-team regularized adjusted plus-minus (RAPM) ratings for NCAA women's college basketball – one row per player-team-season, fit against teammate lineup variation within that team's own possessions (not a cross-team national RAPM). Produced by the sdv-py ncaa_wbb engine's RAPM model on top of the reconstructed lineup/possession data; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_rapm_within_team release tag as csv.gz/parquet/rds.

Loads league-wide regularized adjusted plus-minus (RAPM) ratings for NCAA women's college basketball – one row per player-season, fit against lineup variation across all Division I stints for that season (not a within-team RAPM). Produced by the sdv-py ncaa_wbb engine's league-wide stint-ridge RAPM solver (Torvik-gated to Division I teams) on top of the reconstructed lineup/possession data; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_rapm release tag as csv.gz/parquet/rds.

Loads advanced per-player, per-game box scores for NCAA women's college basketball – one row per player-game, with shooting splits by zone (rim/mid/3), transition and half-court splits, assisted-vs-unassisted splits, and shot-zone efficiency rates. A superset of the traditional box score (points/rebounds/assists/steals/blocks/turnovers/fouls are included alongside the advanced splits). Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_player_box release tag as csv.gz/parquet/rds.

Loads advanced per-team, per-game box scores for NCAA women's college basketball – one row per team-game, with offensive/defensive ratings, shot-zone rates and efficiency (rim/mid/3), rebounding/assist/turnover rates, and the matching opponent (d_*) splits. Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_team_box release tag as csv.gz/parquet/rds.

Loads a lightweight season roster for NCAA women's college basketball – one row per player-team-season with a games-played count. For the fuller roster (jersey, class, position, height, hometown, high school), use load_ncaa_wbb_team_rosters(). Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_rosters release tag as csv.gz/parquet/rds.

Loads the full season roster for NCAA women's college basketball – one row per player-team-season with jersey number, class year, position, height, hometown, high school, and games played/started. Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_team_rosters release tag as csv.gz/parquet/rds.

Loads season schedules/results for NCAA women's college basketball – one row per game with home/away teams and final scores. Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_schedule release tag as csv.gz/parquet/rds.

Loads the season-level stats.ncaa.org team-id / conference crosswalk for NCAA women's college basketball – one row per team-season. Useful for joining ncaa_team_id columns on the other ncaa_wbb_* datasets back to a team name and conference. Produced by the sdv-py ncaa_wbb engine; backed by the ncaa-wbb-hoops-data pipeline, published to the ncaa_wbb_team_ids release tag as csv.gz/parquet/rds.

## Usage

```
load_ncaa_wbb_pbp(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_shots(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_lineups(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_matchup_stints(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_possessions(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_rapm_within_team(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_rapm(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
```

```
)

load_ncaa_wbb_player_box(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_team_box(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_rosters(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_team_rosters(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_schedule(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_ncaa_wbb_team_ids(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit season-ending years for NCAA women's college basketball (e.g. 2024 for the 2023-24 season). Published coverage runs 2010 through the most recent season, with no gaps. Pass seasons = TRUE for every published |

| | |
|---|---|
| | season. (Min: 2010) |
| ... | Additional arguments passed to an underlying function that writes the season data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the data table within the database |

## Details

League-wide regularized adjusted plus-minus – a single ridge fit over all Division I stints per season, distinct from load_ncaa_wbb_rapm_within_team()'s within-team fit. Published coverage runs seasons 2011 through 2026.

## Value

Returns a wehoop_data tibble with one row per play-by-play event.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per shot attempt.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per lineup stint.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per matchup stint.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per possession.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per player-team-season.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per player-season.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per player-game.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per team-game.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per player-team-season.

| col_name | types | description |
|---|---|---|
| season | integer | Season identifier (4-digit season-ending year). |
| team | character | Team name. |
| player | character | Player name (raw, upper-cased). |
| games | integer | Games played (season total). |

Returns a wehoop_data tibble with one row per player-team-season.

See the HTML help or pkgdown reference for the column table.

Returns a `wehoop_data` tibble with one row per game.

| col_name | types | description |
|---|---|---|
| contest_id | character | Unique stats.ncaa.org contest identifier. |
| game_date | character | Date the game was played (MM/DD/YYYY). |
| home | character | Home team name. |
| away | character | Away team name. |
| home_score | integer | Home team score after the event. |
| away_score | integer | Away team score after the event. |
| season | integer | Season identifier (4-digit season-ending year). |

Returns a `wehoop_data` tibble with one row per team-season.

| col_name | types | description |
|---|---|---|
| team | character | Team name. |
| conference | character | Conference affiliation. |
| id | character | stats.ncaa.org team identifier. |
| season | integer | Season identifier (4-digit season-ending year). |

## Author(s)

Saiem Gilani

## Examples

```
try(load_ncaa_wbb_pbp(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_shots(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_lineups(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_matchup_stints(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_possessions(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_rapm_within_team(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_rapm(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_player_box(seasons = most_recent_wbb_season()))
```

```
try(load_ncaa_wbb_team_box(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_rosters(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_team_rosters(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_schedule(seasons = most_recent_wbb_season()))

try(load_ncaa_wbb_team_ids(seasons = most_recent_wbb_season()))
```

---

`load_wbb_group_aliases` **Load conference and division aliases (WBB / WNBA) from the data repo**

---

## Description

Loads every name and id that a source (ESPN, NCAA, WNBA Stats, SDV) uses for a group, with the seasons each alias is valid for. Use it to map a source's conference id or name onto an SDV `group_id`. `load_wbb_group_aliases()` reads the `wbb_groups` release tag and `load_wnba_group_aliases()` the `wnba_groups` tag.

## Usage

```
load_wbb_group_aliases(..., dbConnection = NULL, tablename = NULL)

load_wnba_group_aliases(..., dbConnection = NULL, tablename = NULL)
```

## Arguments

| | |
|---|---|
| `...` | Additional arguments passed to an underlying function that writes the data into a database. |
| `dbConnection` | A `DBIConnection` object, as returned by `DBI::dbConnect()` |
| `tablename` | The name of the data table within the database |

## Value

Returns a `wehoop_data` tibble with one row per alias.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other Conference and Division Group loader functions: `load_wbb_group_seasons()`, `load_wbb_groups()`, `load_wbb_team_group_seasons()`

## Examples

```
try(load_wbb_group_aliases())


try(load_wnba_group_aliases())
```

---

`load_wbb_group_seasons`

**Load conference and division names by season (WBB / WNBA) from the data repo**

---

## Description

Loads one row per group per season it existed, with the group's name, short name, abbreviation, and parent group **as of that season** rather than today's labels, plus its member count. `load_wbb_group_seasons()` reads the `wbb_groups` release tag and `load_wnba_group_seasons()` the `wnba_groups` tag.

## Usage

```
load_wbb_group_seasons(..., dbConnection = NULL, tablename = NULL)

load_wnba_group_seasons(..., dbConnection = NULL, tablename = NULL)
```

## Arguments

| | |
|---|---|
| `...` | Additional arguments passed to an underlying function that writes the data into a database. |
| `dbConnection` | A `DBIConnection` object, as returned by `DBI::dbConnect()` |
| `tablename` | The name of the data table within the database |

## Value

Returns a `wehoop_data` tibble with one row per group-season.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other Conference and Division Group loader functions: load_wbb_group_aliases(), load_wbb_groups(), load_wbb_team_group_seasons()

## Examples

```
try(load_wbb_group_seasons())


try(load_wnba_group_seasons())
```

---

load_wbb_groups **Load conference and division lineages (WBB / WNBA) from the data repo**

---

## Description

Loads one row per group lineage – the league or subdivision, each conference, and each division – with the first and last season each had at least one member. A lineage keeps one group_id across renames that keep continuity (American Athletic to American stays wbb:american); a new body gets a new id. load_wbb_groups() reads the wbb_groups release tag and load_wnba_groups() the wnba_groups tag.

See load_wbb_group_seasons() for per-season names and parents, load_wbb_group_aliases() for the names and ids other sources use, and load_wbb_team_group_seasons() for team membership.

## Usage

```
load_wbb_groups(..., dbConnection = NULL, tablename = NULL)

load_wnba_groups(..., dbConnection = NULL, tablename = NULL)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments passed to an underlying function that writes the data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the data table within the database |

## Value

Returns a wehoop_data tibble with one row per group lineage.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other Conference and Division Group loader functions: `load_wbb_group_aliases()`, `load_wbb_group_seasons()`, `load_wbb_team_group_seasons()`

## Examples

```
try(load_wbb_groups())


try(load_wnba_groups())
```

---

`load_wbb_pbp` **Load cleaned women's college basketball play-by-play from the data repo**

---

## Description

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

`load_wbb_rosters_manifest()` returns the per-season manifest CSV (columns: `season`, `row_count`, `generated_at_utc`, `source_endpoint`) describing which seasons are currently published to the release tag, without downloading any season's full data.

`load_wbb_player_stats_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the player season stats release tag without downloading any season's full data.

`load_wbb_team_stats_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the team season stats release tag without downloading any season's full data.

`load_wbb_standings_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the standings release tag without downloading any season's full data.

`load_wbb_shots_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the shots release tag without downloading any season's full data.

load_wbb_game_rosters_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the game rosters release tag without downloading any season's full data.

load_wbb_officials_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the officials release tag without downloading any season's full data.

Loads season-level team rosters scraped from ESPN. One row per athlete-team-season triple. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_rosters release tag.

Loads season-level player statistics scraped from ESPN. One row per athlete-team-season-statistic-grouping. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_player_season_stats release tag.

Loads season-level team statistics scraped from ESPN. One row per team-season-statistic-grouping. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_team_season_stats release tag.

Loads season-level conference and overall standings scraped from ESPN. One row per team-season. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_standings release tag.

Loads shot events parsed from ESPN women's college basketball play-by-play feeds. One row per shot attempt (made or missed), with court coordinates and shot metadata. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_shots release tag.

Loads per-game rosters scraped from ESPN women's college basketball box scores. One row per athlete-team-game triple, with athlete identifiers, jersey, position, starter flag, and DNP status. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_game_rosters release tag.

Loads game-level officials data scraped from ESPN women's college basketball summary feeds. One row per official-game pair. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_officials release tag.

Loads ESPN WBB athlete core records – identity and biographical fields, one row per athlete who appeared in the season. Backed by the wehoop-wbb-data pipeline that reads raw JSONs from wehoop-wbb-raw and publishes parquet/rds artifacts to the espn_womens_college_basketball_player_core release tag.

This is the only source of athlete bio in the pipeline: the player season stats payload carries no identity at all – not even the athlete id.

Two properties of the source are worth knowing before joining:

- current_team_id is the athlete's CURRENT team, not their team in the requested season. Season team lives in load_wbb_player_box() / load_wbb_player_stats().
- Bio (height / weight / jersey) is a current snapshot that ESPN overwrites in place; it is not era-correct for a historical season. The season dimension here is participation, not the bio's vintage.

Field coverage is era-dependent by nature – headshots exist only for modern players, while college and date of birth thin out the other way.

## Usage

```
load_wbb_pbp(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_team_box(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_player_box(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_schedule(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_rosters_manifest()

load_wbb_player_stats_manifest()

load_wbb_team_stats_manifest()

load_wbb_standings_manifest()

load_wbb_shots_manifest()

load_wbb_game_rosters_manifest()

load_wbb_officials_manifest()

load_wbb_rosters(
  seasons = most_recent_wbb_season(),
```

```
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_player_stats(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_team_stats(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_standings(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_shots(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_game_rosters(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_officials(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_player_core(
```

```
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit years associated with given WBB seasons. (Min: 2004) |
| ... | Additional arguments passed to an underlying function that writes the season data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the player core table within the database |

## Value

A dataframe of play-by-play events with the columns documented below:

See the HTML help or pkgdown reference for the column table.

Returns a tibble

Returns a tibble

Returns a tibble

Returns a wehoop_data tibble with one row per athlete-team-season.

Returns a wehoop_data tibble of player season stats.

Returns a wehoop_data tibble of team season stats.

Columns as documented in the shared basketball_load_wbb_team_stats_schema table.

Returns a wehoop_data tibble of team standings.

Columns as documented in the shared basketball_load_wbb_standings_schema table.

Returns a wehoop_data tibble with one row per shot attempt.

Columns as documented in the shared basketball_load_wbb_shots_schema table.

Returns a wehoop_data tibble with one row per athlete-team-game.

Columns as documented in the shared basketball_load_wbb_game_rosters_schema table.

Returns a wehoop_data tibble with one row per official-game pair.

Columns as documented in the shared basketball_load_wbb_officials_schema table.

Returns a wehoop_data tibble of athlete core records.

## Examples

```
try(load_wbb_pbp())


try(load_wbb_team_box())
```

```
try(load_wbb_player_box())

try(load_wbb_schedule())

try(load_wbb_rosters(seasons = most_recent_wbb_season()))

try(load_wbb_player_stats(seasons = most_recent_wbb_season()))

try(load_wbb_team_stats(seasons = most_recent_wbb_season()))

try(load_wbb_standings(seasons = most_recent_wbb_season()))

try(load_wbb_shots(seasons = most_recent_wbb_season()))

try(load_wbb_game_rosters(seasons = most_recent_wbb_season()))

try(load_wbb_officials(seasons = most_recent_wbb_season()))

try(load_wbb_player_core(seasons = most_recent_wbb_season()))
```

---

`load_wbb_team_group_seasons` **Load team conference membership by season (WBB / WNBA) from the data repo**

---

## Description

Loads one row per team per season with the team's subdivision, conference, and division that season, taken from the most reliable per-season source and cross-checked against a second source where one exists. Membership is never back-filled from today's alignment. `load_wbb_team_group_seasons()` reads the `wbb_groups` release tag and `load_wnba_team_group_seasons()` the `wnba_groups` tag.

## Usage

```
load_wbb_team_group_seasons(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
```

```
)

load_wnba_team_group_seasons(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit seasons: season-ending years for WBB (2025 = 2024-25), calendar years for the WNBA. Published coverage runs from 2002 (WBB) or 1997 (WNBA) through the most recent season. Pass `seasons = TRUE` to read every published season from one file. (Min: 2002 for WBB, 1997 for WNBA) |
| ... | Additional arguments passed to an underlying function that writes the data into a database. |
| dbConnection | A `DBIConnection` object, as returned by `DBI::dbConnect()` |
| tablename | The name of the data table within the database |

## Value

Returns a `wehoop_data` tibble with one row per team-season.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other Conference and Division Group loader functions: `load_wbb_group_aliases()`, `load_wbb_group_seasons()`, `load_wbb_groups()`

## Examples

```
try(load_wbb_team_group_seasons(seasons = most_recent_wbb_season()))


try(load_wnba_team_group_seasons(seasons = most_recent_wnba_season()))
```

load_wnba_pbp **Load cleaned WNBA play-by-play from the data repo**

## Description

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

helper that loads multiple seasons from the data repo either into memory or writes it into a db using some forwarded arguments in the dots

`load_wnba_rosters_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the rosters release tag without downloading any season's full data.

`load_wnba_player_stats_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the player season stats release tag without downloading any season's full data.

`load_wnba_team_stats_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the team season stats release tag without downloading any season's full data.

`load_wnba_standings_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the standings release tag without downloading any season's full data.

`load_wnba_draft_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the draft release tag without downloading any season's full data.

`load_wnba_shots_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the shots release tag without downloading any season's full data.

`load_wnba_game_rosters_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the game rosters release tag without downloading any season's full data.

`load_wnba_officials_manifest()` returns the per-season manifest CSV (`season`, `row_count`, `generated_at_utc`, `source_endpoint`) for the officials release tag without downloading any season's full data.

Loads season-level WNBA team rosters scraped from ESPN. One row per athlete-team-season triple. Backed by the `wehoop-wnba-data` pipeline that reads raw JSONs from `wehoop-wnba-raw` and publishes parquet/rds artifacts to the `espn_wnba_rosters` release tag.

Loads season-level WNBA player statistics scraped from ESPN. One row per athlete-team-season-statistic-grouping. Backed by the `wehoop-wnba-data` pipeline that reads raw JSONs from `wehoop-wnba-raw` and publishes parquet/rds artifacts to the `espn_wnba_player_season_stats` release tag.

Loads season-level WNBA team statistics scraped from ESPN. One row per team-season-statistic-grouping. Backed by the `wehoop-wnba-data` pipeline that reads raw JSONs from `wehoop-wnba-raw` and publishes parquet/rds artifacts to the `espn_wnba_team_season_stats` release tag.

Loads season-level WNBA standings scraped from ESPN. One row per team-season. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_standings release tag.

Loads season-level WNBA draft picks scraped from ESPN. One row per pick. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_draft release tag.

Loads shot events parsed from ESPN WNBA play-by-play feeds. One row per shot attempt (made or missed), with court coordinates and shot metadata. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_shots release tag.

Loads per-game rosters scraped from ESPN WNBA box scores. One row per athlete-team-game triple, with athlete identifiers, jersey, position, starter flag, and DNP status. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_game_rosters release tag.

Loads game-level officials data scraped from ESPN WNBA summary feeds. One row per official-game pair. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_officials release tag.

Loads ESPN WNBA athlete core records – identity and biographical fields, one row per athlete who appeared in the season. Backed by the wehoop-wnba-data pipeline that reads raw JSONs from wehoop-wnba-raw and publishes parquet/rds artifacts to the espn_wnba_player_core release tag.

This is the only source of athlete bio in the pipeline: the player season stats payload carries no identity at all – not even the athlete id.

Two properties of the source are worth knowing before joining:

- current_team_id is the athlete's CURRENT team, not their team in the requested season. Season team lives in load_wnba_player_box() / load_wnba_player_stats().
- Bio (height / weight / jersey) is a current snapshot that ESPN overwrites in place; it is not era-correct for a historical season. The season dimension here is participation, not the bio's vintage.

Field coverage is era-dependent by nature – headshots exist only for modern players, while college and date of birth thin out the other way.

## Usage

```
load_wnba_pbp(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_team_box(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
```

```
)

load_wnba_player_box(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_schedule(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_rosters_manifest()

load_wnba_player_stats_manifest()

load_wnba_team_stats_manifest()

load_wnba_standings_manifest()

load_wnba_draft_manifest()

load_wnba_shots_manifest()

load_wnba_game_rosters_manifest()

load_wnba_officials_manifest()

load_wnba_rosters(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_player_stats(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_team_stats(
  seasons = most_recent_wnba_season(),
```

```
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_standings(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_draft(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_shots(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_game_rosters(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_officials(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_player_core(
  seasons = most_recent_wnba_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit years associated with given WNBA seasons. (Min: 2003) |
| ... | Additional arguments passed to an underlying function that writes the season data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the player core table within the database |

## Value

A dataframe with the columns documented below:

See the HTML help or pkgdown reference for the column table.

Returns a tibble

Returns a tibble

Returns a tibble

Returns a wehoop_data tibble with one row per athlete-team-season.

Returns a wehoop_data tibble of player season stats.

Returns a wehoop_data tibble of team season stats.

Columns as documented in the shared basketball_load_wbb_team_stats_schema table.

Returns a wehoop_data tibble of team standings.

Columns as documented in the shared basketball_load_wbb_standings_schema table.

Returns a wehoop_data tibble of WNBA draft picks.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per shot attempt.

Columns as documented in the shared basketball_load_wbb_shots_schema table.

Returns a wehoop_data tibble with one row per athlete-team-game.

Columns as documented in the shared basketball_load_wbb_game_rosters_schema table.

Returns a wehoop_data tibble with one row per official-game pair.

Columns as documented in the shared basketball_load_wbb_officials_schema table.

Returns a wehoop_data tibble of athlete core records.

## Examples

```
try(load_wnba_pbp())


try(load_wnba_team_box())


try(load_wnba_player_box())


try(load_wnba_schedule())
```

```
try(load_wnba_rosters(seasons = most_recent_wnba_season()))

try(load_wnba_player_stats(seasons = most_recent_wnba_season()))

try(load_wnba_team_stats(seasons = most_recent_wnba_season()))

try(load_wnba_standings(seasons = most_recent_wnba_season()))

try(load_wnba_draft(seasons = most_recent_wnba_season()))

try(load_wnba_shots(seasons = most_recent_wnba_season()))

try(load_wnba_game_rosters(seasons = most_recent_wnba_season()))

try(load_wnba_officials(seasons = most_recent_wnba_season()))

try(load_wnba_player_core(seasons = most_recent_wnba_season()))
```

---

`load_wnba_player_impact` **Load WNBA player-impact ratings (RAPM / SPM / BPM / DARKO) from the data repo**

---

## Description

Loads season-level WNBA player-impact ratings – one row per player-team-season, combining regularized adjusted plus-minus (RAPM), statistical plus-minus (SPM), box plus-minus (BPM), wins above replacement (WAR), and a DARKO-style skill/projection pair. Produced by the sdv-py NBA/WNBA possession-engine model program; published to the `wnba_player_impact` release tag as csv/parquet/rds.

Loads season-level NCAA women's college basketball player-value ratings – one row per player-team-season, with a box-score-derived offensive/defensive/net box plus-minus (BPM). Coverage starts at 2014 (the earliest season with published box-score inputs of sufficient quality for the model); this tag is parquet-only, with no csv/rds sibling assets. Produced by the sdv-py NCAA WBB model program; published to the `wbb_player_value` release tag.

Loads season-level NCAA women's college basketball team ratings – one row per team-season, with adjusted (opponent-strength-normalized) offensive/defensive efficiency, adjusted tempo, raw

efficiency, and a national rank. A KenPom-style adjusted-efficiency-margin rating. This tag is parquet-only, with no csv/rds sibling assets. Produced by the sdv-py NCAA WBB model program; published to the `wbb_ratings` release tag.

## Usage

```
load_wnba_player_impact(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_player_value(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wbb_ratings(
  seasons = most_recent_wbb_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| `seasons` | A vector of 4-digit season-ending years for NCAA women's college basketball. Published coverage runs 2008 through the most recent season, with no gaps. Pass `seasons = TRUE` for every published season. (Min: 2008) |
| `...` | Additional arguments passed to an underlying function that writes the season data into a database. |
| `dbConnection` | A `DBIConnection` object, as returned by `DBI::dbConnect()` |
| `tablename` | The name of the data table within the database |

## Value

Returns a `wehoop_data` tibble with one row per player-team-season.

See the HTML help or pkgdown reference for the column table.

Returns a `wehoop_data` tibble with one row per player-team-season.

See the HTML help or pkgdown reference for the column table.

Returns a `wehoop_data` tibble with one row per team-season.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## Examples

```
try(load_wnba_player_impact(seasons = most_recent_wnba_stats_season()))

try(load_wbb_player_value(seasons = most_recent_wbb_season()))

try(load_wbb_ratings(seasons = most_recent_wbb_season()))
```

---

load_wnba_stats_leaguedash

# Load wehoop WNBA Stats League Dashboard cube

---

## Description

Loads one asset of the wnba_stats_leaguedash release tag – the Python-scraped leaguedash* parameter cube that supersedes the old R-scraped player / team / lineups / standings tags. The cube publishes 24 tables per season and is the full surface behind the compatibility reshapes in load_wnba_stats_player_stats(), load_wnba_stats_team_stats(), load_wnba_stats_lineups() and load_wnba_stats_standings(); several tables (player_bio, *_master, team_stats_fourfactors, the non-5-man and non-Base/Advanced lineups_* measures) are reachable only through this function.

## Usage

```
load_wnba_stats_leaguedash(
  seasons = most_recent_wnba_stats_season(),
  table = NULL,
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit years associated with given WNBA seasons. Published coverage runs 1997 through the most recent season, with no gaps. Pass seasons = TRUE for every published season. (Min: 1997) |
| table | Name of the cube table to load. One of player_bio, player_master, player_stats_base, player_stats_advanced, player_stats_misc, player_stats_scoring, player_stats_usage, player_stats_defense, team_master, team_stats_base, team_stats_advanced, team_stats_misc, team_stats_scoring, team_stats_defense, team_stats_opponent, |

| | |
|---|---|
| | team_stats_fourfactors, lineups_master, lineups_base, lineups_advanced, lineups_misc, lineups_scoring, lineups_opponent, lineups_fourfactors, standings. |
| ... | Additional arguments passed to an underlying function that writes the season data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the data table within the database |

## Value

Returns a wehoop_data tibble of the requested cube table, one row per player-season, team-season or lineup-season depending on table. Column sets differ per table; the *_master tables are wide joins of every measure type for that entity.

## See Also

Other WNBA Stats loader functions: load_wnba_stats_rosters()

## Examples

```
try(load_wnba_stats_leaguedash(seasons = most_recent_wnba_stats_season(),
                               table = "player_bio"))
```

---

load_wnba_stats_rosters

**Load cleaned WNBA Stats API season rosters from the data repo**

---

## Description

Loads season-level WNBA team rosters scraped from the WNBA Stats API. One row per athlete-team-season triple. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_rosters release tag.

Loads season-level WNBA coaching staff data scraped from the WNBA Stats API. One row per coach-team-season triple. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_coaches release tag.

**[Deprecated]** Loads season-level WNBA player statistics. **Deprecated**: the wnba_stats_player_season_stats release tag (R-scraped, Base/Advanced/Misc/Scoring/Usage/ Defense measures) is superseded by the wnba_stats_leaguedash tag (Python-scraped parameter cube, same 6 measure types plus player_bio and a wide player_master mega). This function reshapes the cube back into the old stacked-by-measure_type contract for compatibility; call the cube's player_stats_*/player_master assets directly with load_wnba_stats_leaguedash() for the full surface.

**[Deprecated]** Loads season-level WNBA 5-man lineup statistics (leaguedashlineups-style outputs). **Deprecated**: the wnba_stats_lineups release tag (R-scraped, Base/Advanced measures,

5-man only) is superseded by the wnba_stats_leaguedash tag (Python-scraped parameter cube: 6 measure types x 2/3/4/5-man). This function reshapes the cube back into the old 5-man Base+Advanced contract for compatibility; call the cube's lineups_* / lineups_master assets directly with load_wnba_stats_leaguedash for the full surface.

**[Deprecated]** Loads season-level team statistics (leaguedashteamstats-style outputs). **Deprecated**: the wnba_stats_team_season_stats release tag (R-scraped, Base/ Advanced/Misc/Scoring/Defense/Opponent measures) is superseded by the wnba_stats_leaguedash tag (Python-scraped parameter cube, same 6 measures plus Four Factors and a wide team_master mega). This function reshapes the cube back into the old stacked-by-measure_type contract for compatibility; call the cube's team_stats_* / team_master assets directly with load_wnba_stats_leaguedash() for the full surface.

**[Deprecated]** Loads season-level WNBA standings (leaguestandingsv3-style outputs). One row per team-season. **Deprecated**: the wnba_stats_standings release tag (R-scraped) is superseded by the wnba_stats_leaguedash tag's standings asset (same underlying endpoint/params, Python-scraped) — this is close to a pure passthrough.

Loads season-level WNBA draft picks scraped from the WNBA Stats API (drafthistory-style outputs). One row per pick. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_draft release tag.

Loads shot events scraped from the WNBA Stats API. One row per shot attempt with legacy court coordinates, action/sub type, distance, and made/missed result, carried through from the V3 play-by-play feed (not shotchartdetail). Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_shots release tag.

Loads the per-game **inactive player** list scraped from the WNBA Stats API – the InactivePlayers result set of boxscoresummaryv2. One row per inactive athlete-game pair, not a full per-game roster: use load_wnba_stats_player_game_logs() for the athletes who did play. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_game_rosters release tag.

Coverage is thin by construction because the upstream result set is thin: stats.wnba.com reports inactives for only a fraction of games, and most heavily for 2006-2008 and 2021 onward. Season row counts range from 1 (1997) to 610 across 184 games (2026); 2009-2012 carry fewer than 10 rows each. A season with almost no rows is upstream sparsity, not a download failure.

Loads game-level officials data scraped from the WNBA Stats API (boxscoresummaryv2-style outputs). One row per official-game pair. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs from wehoop-wnba-stats-raw and publishes parquet/rds artifacts to the wnba_stats_officials release tag.

Loads per-player per-game logs scraped from stats.wnba.com/leaguegamelog?PlayerOrTeam=P (one row per athlete-game pair: minutes, shooting splits, rebounds, steals, blocks, turnovers, personal fouls, plus/minus). Backed by the wehoop-wnba-stats-data pipeline that publishes parquet/rds artifacts to the wnba_stats_player_game_logs release tag.

Loads season-level WNBA schedules scraped from stats.wnba.com/leaguegamefinder (regular season + playoffs combined, pre-rejoined home/away). Backed by the wehoop-wnba-stats-data pipeline that publishes parquet/rds artifacts to the wnba_stats_schedules release tag.

Loads season-level WNBA play-by-play (V3), supplied by wnba_pbp() in the upstream compile

script. Backed by the wehoop-wnba-stats-data pipeline that publishes parquet/rds artifacts to the wnba_stats_pbp release tag.

load_wnba_stats_rosters_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats rosters release tag without downloading any season's full data.

load_wnba_stats_coaches_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats coaches release tag without downloading any season's full data.

load_wnba_stats_player_stats_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats player season stats release tag without downloading any season's full data.

load_wnba_stats_lineups_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats lineups release tag without downloading any season's full data.

load_wnba_stats_team_stats_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats team season stats release tag without downloading any season's full data.

load_wnba_stats_standings_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats standings release tag without downloading any season's full data.

load_wnba_stats_draft_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats draft release tag without downloading any season's full data.

load_wnba_stats_shots_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats shots release tag without downloading any season's full data.

load_wnba_stats_game_rosters_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats game rosters release tag without downloading any season's full data.

load_wnba_stats_officials_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats officials release tag without downloading any season's full data.

load_wnba_stats_player_game_logs_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats player game logs release tag without downloading any season's full data.

load_wnba_stats_schedule_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats schedules release tag without downloading any season's full data.

load_wnba_stats_pbp_manifest() returns the per-season manifest CSV (season, row_count, generated_at_utc, source_endpoint) for the WNBA Stats play-by-play release tag without downloading any season's full data.

Loads possession-level data derived from the WNBA Stats API play-by-play – one row per possession, with the on-court 5-man lineups for both teams, shooting/rebounding/turnover splits, and the possession start type. Backed by the wehoop-wnba-stats-data pipeline that reads raw JSONs

from wehoop-wnba-stats-raw and publishes csv.gz/parquet/rds artifacts to the wnba_stats_possessions release tag.

## Usage

```
load_wnba_stats_rosters(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_coaches(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_player_stats(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_lineups(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_team_stats(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_standings(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_draft(
  seasons = most_recent_wnba_stats_season(),
```

```
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_shots(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_game_rosters(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_officials(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_player_game_logs(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_schedule(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_pbp(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)

load_wnba_stats_rosters_manifest()
```

```
load_wnba_stats_coaches_manifest()

load_wnba_stats_player_stats_manifest()

load_wnba_stats_lineups_manifest()

load_wnba_stats_team_stats_manifest()

load_wnba_stats_standings_manifest()

load_wnba_stats_draft_manifest()

load_wnba_stats_shots_manifest()

load_wnba_stats_game_rosters_manifest()

load_wnba_stats_officials_manifest()

load_wnba_stats_player_game_logs_manifest()

load_wnba_stats_schedule_manifest()

load_wnba_stats_pbp_manifest()

load_wnba_stats_possessions(
  seasons = most_recent_wnba_stats_season(),
  ...,
  dbConnection = NULL,
  tablename = NULL
)
```

## Arguments

| | |
|---|---|
| seasons | A vector of 4-digit years associated with given WNBA seasons. Published coverage runs 1997 through the most recent season, with no gaps. Pass seasons = TRUE for every published season. (Min: 1997) |
| ... | Additional arguments passed to an underlying function that writes the season data into a database. |
| dbConnection | A DBIConnection object, as returned by DBI::dbConnect() |
| tablename | The name of the data table within the database |

## Value

Returns a wehoop_data tibble with one row per athlete-team-season.

Returns a wehoop_data tibble with one row per coach-team-season.

Returns a wehoop_data tibble of player season stats.

Returns a wehoop_data tibble of lineup stats.

Returns a wehoop_data tibble of team season stats.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble of team standings.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble of WNBA draft picks.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per shot attempt.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per inactive athlete-game pair.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble with one row per official-game pair.

| col_name | types | description |
|---|---|---|
| official_id | integer | Unique official / referee identifier. |
| first_name | character | Official's first name. |
| last_name | character | Official's last name. |
| jersey_num | character | Jersey number worn by the official. |
| season | integer | Season identifier (4-digit year). |
| game_id | character | Unique game identifier. |

Returns a wehoop_data tibble of per-athlete per-game log rows.

Returns a wehoop_data tibble of per-season schedules, one row per game with the home/away sides pre-joined into home_* / away_* columns.

See the HTML help or pkgdown reference for the column table.

Returns a wehoop_data tibble of per-event play-by-play rows.

Returns a wehoop_data tibble with one row per possession.

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Stats loader functions: load_wnba_stats_leaguedash()

## Examples

```
try(load_wnba_stats_rosters(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_coaches(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_player_stats(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_lineups(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_team_stats(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_standings(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_draft(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_shots(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_game_rosters(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_officials(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_player_game_logs(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_schedule(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_pbp(seasons = most_recent_wnba_stats_season()))

try(load_wnba_stats_possessions(seasons = most_recent_wnba_stats_season()))
```

---

`load_wnba_team_crosswalk`

# Load cached WNBA team crosswalk from the data repo

---

## Description

Loads cached team-crosswalk snapshots that map ESPN team identifiers to WNBA Stats API team identifiers. One row per team-season. Backed by the `wnba_crosswalk` release tag in `sportsdataverse-data`.

Loads cached schedule-crosswalk snapshots that map ESPN game identifiers to WNBA Stats API game identifiers. One row per game-season. Backed by the `wnba_crosswalk` release tag in `sportsdataverse-data`.

Loads cached player-crosswalk snapshots that map ESPN athlete identifiers to WNBA Stats API player identifiers. One row per athlete-season. Backed by the `wnba_crosswalk` release tag in `sportsdataverse-data`.

Loads cached team-crosswalk snapshots that map ESPN team identifiers to Fox Sports and Bart Torvik team identifiers for Women's College Basketball. One row per team-season. Backed by the `wbb_crosswalk` release tag in `sportsdataverse-data`.

Loads cached schedule-crosswalk snapshots that map ESPN game identifiers to Bart Torvik game identifiers for Women's College Basketball. One row per game-season. Backed by the `wbb_crosswalk` release tag in `sportsdataverse-data`.

Loads cached player-crosswalk snapshots that map ESPN athlete identifiers to Fox Sports player identifiers for Women's College Basketball. One row per athlete-season. Backed by the `wbb_crosswalk` release tag in `sportsdataverse-data`.

## Usage

```
load_wnba_team_crosswalk(seasons = most_recent_wnba_season())

load_wnba_schedule_crosswalk(seasons = most_recent_wnba_season())

load_wnba_player_crosswalk(seasons = most_recent_wnba_season())

load_wbb_team_crosswalk(seasons = most_recent_wbb_season())

load_wbb_schedule_crosswalk(seasons = most_recent_wbb_season())

load_wbb_player_crosswalk(seasons = most_recent_wbb_season())
```

## Arguments

| | |
|---|---|
| `seasons` | A vector of 4-digit years associated with given WBB seasons. (Min: 2014) |

## Value

A `wehoop_data` tibble of cached team-crosswalk rows.

A `wehoop_data` tibble of cached schedule-crosswalk rows.

A `wehoop_data` tibble of cached player-crosswalk rows.

A `wehoop_data` tibble of cached WBB team-crosswalk rows.

A `wehoop_data` tibble of cached WBB schedule-crosswalk rows.

A `wehoop_data` tibble of cached WBB player-crosswalk rows.

## See Also

Other WNBA Crosswalk Functions: `wnba_player_crosswalk()`, `wnba_schedule_crosswalk()`, `wnba_team_crosswalk()`

Other WBB Crosswalk Functions: `wbb_player_crosswalk()`, `wbb_schedule_crosswalk()`, `wbb_team_crosswalk()`

## Examples

```
try(load_wnba_team_crosswalk(seasons = most_recent_wnba_season()))

try(load_wnba_schedule_crosswalk(seasons = most_recent_wnba_season()))

try(load_wnba_player_crosswalk(seasons = most_recent_wnba_season()))

try(load_wbb_team_crosswalk(seasons = most_recent_wbb_season()))

try(load_wbb_schedule_crosswalk(seasons = most_recent_wbb_season()))

try(load_wbb_player_crosswalk(seasons = most_recent_wbb_season()))
```

---

# lpb_competition — LPB / LPBF Liga Profesional de Baloncesto (Panama) (Atrium shortcuts)

---

## Description

**Per-competition shortcuts over the Atrium embed API for LPB / LPBF Liga Profesional de Baloncesto (Panama) (`website_id = 22`; men's and women's).**

Thin bindings of the parameterised `atrium_*()` wrappers to this site, minted from the competition registry (sdv-py `tools/codegen/basketball_competitions.yaml`); the full surface stays on the `atrium_*()` wrappers.

## Usage

```
lpb_seasons(locale = NULL)
lpb_schedule(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lpb_standings(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lpb_teams(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
lpb_player_box(fixture_id, locale = NULL)
lpb_team_box(fixture_id, locale = NULL)
lpb_pbp(fixture_id, locale = NULL)
lpb_shots(fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| `season` | (*integer* optional): Season END year (season rule below). Omit `season` and `season_id` for the site's current season. |
| `season_id` | (*character* optional): Season UUID (`season_id` in the seasons listing); skips the END-year lookup. |

`competition_id` (*character* optional): Competition UUID (a site carries many: conferences, cups); needed when several share the season.

`locale` (*character* optional): Label locale, e.g. `"en-EN"`; omitted = the site's own language.

`fixture_id` (*character* required): Fixture UUID (`fixture_id` in the schedule).

## Details

Season rule: calendar year.

Errors and terms of use are those of `atrium_fixtures()`: unofficial use of the public widget endpoint, at least 1 s between requests, not republished as release assets. The site also serves women's competitions (pass their `competition_id`).

```
lpb_seasons()
lpb_schedule()
```

## Value

A `wehoop_data` tibble with the columns of the base wrapper, one table per function:

`lpb_seasons()`:

The columns of `atrium_seasons()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, |
| season_id | character | Season UUID; pass it as season_id to any season-scoped wrapper. |
| competition_id | character | Competition UUID; pass it as competition_id to pick one of several seasons of an END year. |
| competition_name | character | Competition name in the site's language; the same name can appear for several years. |
| season_name | character | Season name in the site's language, e.g. Betclic ELITE 2025. |
| abbreviation_local | character | Local abbreviation of the season; null in every body seen. |
| year | integer | The site's own year of the season: the START year on LNB (site 12), the calendar year elsew |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year on th |
| is_default_season | logical | True for the season the body is about (the site's default when none was asked). |

`lpb_schedule()`:

The columns of `atrium_fixtures()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| season_id | character | Season UUID the page is about (the site's default season when none was asked). |
| competition_id | character | Competition UUID of the season. |
| competition_name | character | Competition name in the site's language, e.g. Betclic ELITE or LIGA CALIENTE.M |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar yea |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| name | character | Fixture name as the site prints it (e.g. Paris - Lyon-Villeurbanne); null on LNBP. |
| round | character | Round of the game as the site words it, e.g. Journee: 30. |

| | | |
|---|---|---|
| round_code | character | Round code; null in every body seen. |
| pool | character | Pool the game belongs to; null when the competition has none. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a s |
| stage_code | character | Opaque 30-hex stage code of the game; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (YYYY-MM-DDTHH:MM:SS, |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_loc |
| times_unconfirmed | logical | True while the start time is not final. |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_place | integer | Result place of the home team: 1 for the winner, 2 for the loser. |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_draw | logical | True when the game ended level for the home team. |
| home_shootout_attempts | character | Shootout attempts of the home team as sent (an empty string becomes null). |
| home_secondary_score | character | Secondary score of the home team (unused in the samples; shown as text). |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_place | integer | Result place of the away team: 1 for the winner, 2 for the loser. |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_draw | logical | True when the game ended level for the away team. |
| away_shootout_attempts | character | Shootout attempts of the away team as sent (an empty string becomes null). |
| away_secondary_score | character | Secondary score of the away team (unused in the samples; shown as text). |

`lpb_standings()`:

The columns of `atrium_standings()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year o |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |

| | | |
|---|---|---|
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

`lpb_teams()`:

The columns of `atrium_teams()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Au |
| season_id | character | Season UUID of the team list. |
| entity_id | character | Team UUID (entity): one id per club across seasons, shared by every site that shows the team. |
| name | character | Team name in the site's language. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

`lpb_player_box()`:

The columns of `atrium_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID the player is rostered with for this game. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| person_name | character | Player name as the site prints it (upper case on LNBP and NBL1). |
| bib | character | Jersey number the player wears, as text. |
| position | character | Position code on the roster (LNB, e.g. PG); null on LNBP. |
| starter | logical | True when the player started the game. |
| active | logical | True when the player is on the court in a live game. |
| participated | logical | True when the player entered the game; false rows have null statistics. |
| did_not_play_reason | character | Reason the player did not play; null in every body seen. |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |

| | | |
|---|---|---|
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, this game). |
| points | integer | Points scored (player, this game). |
| rebounds | integer | Total rebounds (player, this game). |
| rebounds_offensive | integer | Offensive rebounds (player, this game). |
| rebounds_defensive | integer | Defensive rebounds (player, this game). |
| assists | integer | Assists (player, this game). |
| steals | integer | Steals (player, this game). |
| blocks | integer | Shots blocked (player, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, this game). |
| turnovers | integer | Turnovers (player, this game). |
| fouls_total | integer | Fouls committed (player, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, this game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) ( |
| efficiency | integer | Efficiency rating, from the site's own formula (player, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, this game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, this game). |
| points_two_made | integer | Two-point field goals made (player, this game). |
| points_two_attempted | integer | Two-point field goals attempted (player, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, this game). |
| points_three_made | integer | Three-point field goals made (player, this game). |
| points_three_attempted | integer | Three-point field goals attempted (player, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, this game). |
| free_throws_made | integer | Free throws made (player, this game). |
| free_throws_attempted | integer | Free throws attempted (player, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, this game). |

`lpb_team_box()`:

The columns of `atrium_team_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 2 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail route |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID (from the fixture's isHome flag). |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome fla |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its pla |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (team, this game). |
| points | integer | Points scored (team, this game). |
| rebounds | integer | Total rebounds (team, this game). |
| rebounds_offensive | integer | Offensive rebounds (team, this game). |
| rebounds_defensive | integer | Defensive rebounds (team, this game). |
| assists | integer | Assists (team, this game). |
| steals | integer | Steals (team, this game). |
| blocks | integer | Shots blocked (team, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (team, this game). |

| | | |
|---|---|---|
| turnovers | integer | Turnovers (team, this game). |
| fouls_total | integer | Fouls committed (team, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (team, this game |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lin |
| efficiency | integer | Efficiency rating, from the site's own formula (team, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (team, this game |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (team, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, this game). |
| points_two_made | integer | Two-point field goals made (team, this game). |
| points_two_attempted | integer | Two-point field goals attempted (team, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (team, this game). |
| points_three_made | integer | Three-point field goals made (team, this game). |
| points_three_attempted | integer | Three-point field goals attempted (team, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (team, this game). |
| free_throws_made | integer | Free throws made (team, this game). |
| free_throws_attempted | integer | Free throws attempted (team, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (team, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (team, this game). |
| extra_fouls_total | integer | Fouls committed (team, credited to the team rather than a player). |
| extra_rebounds | integer | Total rebounds (team, credited to the team rather than a player). |
| extra_rebounds_defensive | integer | Defensive rebounds (team, credited to the team rather than a player). |
| extra_rebounds_offensive | integer | Offensive rebounds (team, credited to the team rather than a player). |
| extra_turnovers | integer | Turnovers (team, credited to the team rather than a player). |
| total_biggest_lead | integer | Biggest lead of the game, in points (team, game total). |
| total_biggest_scoring_run | integer | Longest unanswered scoring run of the game, in points (team, game total). |
| total_field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, game total). |
| total_free_throws_percentage | double | Free-throw percentage, 0-100 (team, game total). |
| total_lead_changes | integer | Number of lead changes in the game (team, game total). |
| total_points_fast_break | integer | Fast-break points (LNBP only) (team, game total). |
| total_points_from_bench | integer | Points scored by non-starters (LNBP only) (team, game total). |
| total_points_from_turnover | integer | Points scored off opponent turnovers (LNBP only) (team, game total). |
| total_time_in_lead | character | Time the team led, as an ISO 8601 duration (PT36M11S) (team, game total). |
| total_time_in_lead_seconds | integer | Time the team led, in whole seconds (team, game total). |
| total_turnovers | integer | Turnovers (team, game total). |

`lpb_pbp()`:

The columns of `atrium_pbp()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the event in the game (periods in order, events chronological). |
| event_id | character | Event UUID, shared with the same event in atrium_shots. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |

| | | |
|---|---|---|
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID; null for team events (time-outs, team rebounds). |
| person_name | character | Player the event is credited to; null for team events (time-outs, team rebounds). |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when a shot or free throw was made, false when missed; null for other events. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

`lpb_shots()`:

The columns of `atrium_shots()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the shot among the shots of the game (not the pbp sequence). |
| event_id | character | Event UUID of the shot; the same event in atrium_pbp. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID of the shooter. |
| person_name | character | Name of the shooter. |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when the shot was made, false when missed. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,

atrium_player_stats(), atrium_players(), atrium_season_id(), atrium_seasons(), atrium_shots(), atrium_standings(), atrium_standings_summary(), atrium_team(), atrium_team_box_score(), atrium_team_stats(), atrium_teams(), lmb_competition, lnbp_competition, nbl1_competition

---

## most_recent_wbb_season

*Most Recent Women's College Basketball Season*

---

### Description

Returns the most recent women's college basketball season year as an integer based on the current system date. The NCAA WBB season spans two calendar years; this helper rolls over to the next season starting in October. For example, calls made on 2025-11-15 return `2026`, while calls made on 2025-03-15 return `2025`.

### Usage

```
most_recent_wbb_season()
```

### Value

An integer giving the season-ending year (e.g. `2025` for the 2024-25 season).

### Examples

```
most_recent_wbb_season()
```

---

## most_recent_wnba_season

*Most Recent WNBA Season*

---

### Description

Returns the most recent WNBA season year as an integer based on the current system date. The WNBA season runs from May through October within a single calendar year; this helper rolls forward to the new season starting in May.

### Usage

```
most_recent_wnba_season()
```

### Value

An integer giving the WNBA season year (e.g. `2025`).

### Examples

```
most_recent_wnba_season()
```

## most_recent_wnba_stats_season *Most Recent WNBA Stats API Season*

### Description

Returns the most recent WNBA Stats API season year as an integer. This is a thin wrapper around `most_recent_wnba_season()` kept as a separate helper for naming symmetry with the `wnba_stats_*` family of loaders (mirrors the `most_recent_*_season()` convention used elsewhere in the package). The WNBA Stats API and ESPN's WNBA endpoints share the same calendar-year season identifier, so the returned value is identical to `most_recent_wnba_season()`.

### Usage

```
most_recent_wnba_stats_season()
```

### Value

An integer giving the WNBA season year (e.g. `2025`).

### Examples

```
most_recent_wnba_stats_season()
```

## nbl_rosetta_match **NBL Rosetta Match**

### Description

One match's header: status, scores, clock, venue, season and teams. Box score and play-by-play: the `match_*` frames.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}/live/all`

### Usage

```
nbl_rosetta_match(match_id)
```

### Arguments

| | |
|---|---|
| `match_id` | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competit |
| match_id | character | DataCore fixture id (UUID v1); equal to `id`. |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box sc |
| start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| match_type | character | Match type: regular, final (FINAL upper case seen) |
| attendance | integer | Attendance, a STRING; null when not recorded. |
| clock | character | Game clock of the latest event, HH:MM:SS remaining in the period. |
| current_period | integer | Latest period number as a STRING; null on older matches. |
| clock_running | logical | Game clock is running. |
| home_score | integer | Home final score, a STRING; null before tip-off. |
| away_score | integer | Away final score, a STRING; null before tip-off. |
| round | character | Round label: a number as a string in the regular season ("13"), words in the f |
| start_time | character | Tip-off, UTC, with no offset or a Z suffix (ESPN dates the same Perth v SE M |
| status | character | Provider fixture status: CONFIRMED, SCHEDULED. |

| | | |
|---|---|---|
| external_media_id | character | Media id. Through NBL25 (2024-25): the Genius Sports match id (7 digits, e |
| match_slug | character | URL slug of the match page on nbl.com.au. |
| match_title | character | Match title ("<home> v <away> <men/women> DD-MM-YYYY") |
| secondary_external_id | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| live_exclusive | logical | Match is a live-stream exclusive. |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and t |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhi |
| season_competition_id | character | Rosetta competition id (UUID v4) |
| season_competition_name | character | Name of the competition the season belongs to (e.g. NBL, WNBL, NBL1 No |
| season_competition_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lo |
| season_competition_league_name | character | Name of the league the competition belongs to (NBL, WNBL or NBL1). |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| venue_id | character | DataCore id (UUID v1, Sportradar) of the venue; stable across seasons and tl |
| rosetta_venue_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the venue; routes that take this e |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_code | character | Venue code; often null. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_suburb | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Country; often null. |
| home_team_id | character | DataCore id (UUID v1, Sportradar) of the home team; stable across seasons a |
| home_team_name | character | Display name of the team. |
| home_team_external_team_name | character | Team name as DataCore spells it. |
| home_team_code | character | Team code of the home team, e.g. PER or SYD. |
| away_team_id | character | DataCore id (UUID v1, Sportradar) of the away team; stable across seasons a |
| away_team_name | character | Display name of the team. |
| away_team_external_team_name | character | Team name as DataCore spells it. |
| away_team_code | character | Team code of the away team, e.g. PER or SYD. |
| data_source | character | Backend that built the body: nexus-api or lambda-cache (observed). It varies |
| match_status_string | character | Status banner text, e.g. FINAL. |
| play_by_play_updated_datetime | character | Last play-by-play update, ISO 8601 UTC. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_`
`nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`,
`nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`,
`nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`,

nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_match(match_id = "b217a699-4bef-11f0-bc08-933162362355"))
```

---

`nbl_rosetta_match_by_id`

# NBL Rosetta Match by Id

---

## Description

One match's header (the same body as nbl_rosetta_match): status, scores, venue, season and teams.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}`

## Usage

```
nbl_rosetta_match_by_id(match_id)
```

## Arguments

| | |
|---|---|
| `match_id` | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.

- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_match_by_id(match_id = "b217a699-4bef-11f0-bc08-933162362355")
```

## Value

A `wehoop_data` tibble (one row):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competit |
| match_id | character | DataCore fixture id (UUID v1); equal to `id`. |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box sc |
| start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| match_type | character | Match type: regular, final (FINAL upper case seen) |
| attendance | integer | Attendance, a STRING; null when not recorded. |
| clock | character | Game clock of the latest event, HH:MM:SS remaining in the period. |
| current_period | integer | Latest period number as a STRING; null on older matches. |
| clock_running | logical | Game clock is running. |
| home_score | integer | Home final score, a STRING; null before tip-off. |
| away_score | integer | Away final score, a STRING; null before tip-off. |
| round | character | Round label: a number as a string in the regular season ("13"), words in the f |
| start_time | character | Tip-off, UTC, with no offset or a Z suffix (ESPN dates the same Perth v SE M |
| status | character | Provider fixture status: CONFIRMED, SCHEDULED. |
| external_media_id | character | Media id. Through NBL25 (2024-25): the Genius Sports match id (7 digits, e |
| match_slug | character | URL slug of the match page on nbl.com.au. |
| match_title | character | Match title ("<home> v <away> <men/women> DD-MM-YYYY") |
| secondary_external_id | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| live_exclusive | logical | Match is a live-stream exclusive. |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and t |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhi |
| season_competition_id | character | Rosetta competition id (UUID v4) |
| season_competition_name | character | Name of the competition the season belongs to (e.g. NBL, WNBL, NBL1 No |
| season_competition_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lo |
| season_competition_league_name | character | Name of the league the competition belongs to (NBL, WNBL or NBL1). |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| venue_id | character | DataCore id (UUID v1, Sportradar) of the venue; stable across seasons and th |
| rosetta_venue_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the venue; routes that take this e |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |

| | | |
|---|---|---|
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_code | character | Venue code; often null. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_suburb | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Country; often null. |
| home_team_id | character | DataCore id (UUID v1, Sportradar) of the home team; stable across seasons a |
| home_team_name | character | Display name of the team. |
| home_team_external_team_name | character | Team name as DataCore spells it. |
| home_team_code | character | Team code of the home team, e.g. PER or SYD. |
| away_team_id | character | DataCore id (UUID v1, Sportradar) of the away team; stable across seasons a |
| away_team_name | character | Display name of the team. |
| away_team_external_team_name | character | Team name as DataCore spells it. |
| away_team_code | character | Team code of the away team, e.g. PER or SYD. |
| data_source | character | Backend that built the body: nexus-api or lambda-cache (observed). It varies |
| match_status_string | character | Status banner text, e.g. FINAL. |
| play_by_play_updated_datetime | character | Last play-by-play update, ISO 8601 UTC. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`, `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

---

# `nbl_rosetta_match_pbp` NBL Rosetta Match Play-by-Play

---

## Description

One match's play-by-play with shot x/y, oldest first (NBL 2025-26 included, which ESPN's nbl feed lacks).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}/live/all`

## Usage

```
nbl_rosetta_match_pbp(match_id)
```

## Arguments

| | |
|---|---|
| match_id | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

## Value

A wehoop_data tibble (one row per event, sorted on action_id):

| col_name | type | description |
|---|---|---|
| match_id | character | DataCore fixture id of the match (UUID v1): the {match_id} route key and fixtureId in p |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-by-play rows; r |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box score and play |
| action_id | integer | Event sequence number within the match (ascending in feed time) |
| period | integer | Period number, an INTEGER here (1-4 = quarters) |
| period_type | character | Period type: regular (overtime unobserved) |
| clock | character | Game clock at the event, MM:SS remaining in the period; null on some feed-control even |
| clock_seconds | double | Game clock at the event as seconds (parsed from clock, MM:SS); null for events without a |
| shot_clock | character | Shot clock at the event, MM:SS; in Genius-era matches it repeats the game clock (no shot |

| | | |
|---|---|---|
| shot_clock_seconds | double | Shot clock at the event as seconds (parsed from shot_clock); in Genius-era backfills it rep |
| score_home | integer | Home team's running score after the event (score_1 in the body). |
| score_away | integer | Away team's running score after the event (score_2 in the body). |
| action_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, t |
| sub_type | character | Event detail (shot type, foul type, "1Of2" free throw, In/Out...); absent when the event has |
| sub_type_key | character | sub_type folded to snake_case, so the Title Case 2025-26 spelling and the camelCase bacl |
| success | logical | Shot or free throw made (true) or missed (false). DataCore-era feeds (events with fixtureId |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to join on |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| is_home | logical | True when the event's team is the match's home team; null when the event has no team. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the id to join |
| player_name | character | Full name of the player the event belongs to; null for events no player owns (team rebound |
| jersey_number | character | Jersey number of the player at the event, as text (a WNBL backfill writes 6.0). |
| x | double | Shot or event position along the court length, percent 0-100, in game orientation (each tea |
| y | double | Shot or event position across the court width, percent 0-100 (free throws and close shots c |
| readable_action_type | character | Human-readable event label, e.g. "Game - Game completed". |
| system_action_type | character | Machine event label, e.g. game_game_completed. |
| timestamp | character | Wall-clock time of the event, ISO 8601 without offset (UTC) |
| feed | character | Feed shape: datacore (NBL and WNBL 2025-26: events carry fixtureId, arrive newest firs |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`, `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

## Examples

```
try(nbl_rosetta_match_pbp(match_id = "b217a699-4bef-11f0-bc08-933162362355"))
```

---

`nbl_rosetta_match_player_box`

# NBL Rosetta Match Player Box Score

---

## Description

One match's official player box score, one row per player line (period 0 = whole game).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}/live/all`

## Usage

```
nbl_rosetta_match_player_box(match_id)
```

## Arguments

| | |
|---|---|
| match_id | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per player line):

| col_name | type | description |
|---|---|---|
| match_id | character | DataCore fixture id of the match (UUID v1): the {match_id} route key and fixtureId i |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box score and p |
| stat_line_id | character | Provider record hash of this statistics row (64 hex characters) |
| rosetta_stat_line_id | character | Rosetta id of this statistics row (UUID v4) |
| assists | integer | Assists in the match, or in the period on period rows; 0 where the era did not record it ( |

| | | |
|---|---|---|
| blocks | integer | Blocks in the match, or in the period on period rows; 0 where the era did not record it ( |
| defensive_rebounds | integer | Defensive rebounds in the match, or in the period on period rows; 0 where the era did r |
| efficiency | integer | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - missed FT - TOV) i |
| field_goals_attempted | integer | Field goals attempted in the match, or in the period on period rows; 0 where the era did |
| field_goals_made | integer | Field goals made in the match, or in the period on period rows; 0 where the era did not |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls | integer | Fouls committed (personal + technical + others) in the match, or in the period on perio |
| free_throws_attempted | integer | Free throws attempted in the match, or in the period on period rows; 0 where the era di |
| free_throws_made | integer | Free throws made in the match, or in the period on period rows; 0 where the era did no |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| jersey_number | character | Jersey number, a STRING. |
| minutes | double | Minutes played: "MM:SS" string in player_match_statistics, decimal minutes in calcul |
| minutes_string | character | Minutes played as an ISO 8601 duration (PT30M32S); null on NBL1. |
| offensive_rebounds | integer | Offensive rebounds in the match, or in the period on period rows; 0 where the era did n |
| participated | logical | Player took part in the game. |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = first/second/ |
| period_type | character | Period type: regular (overtime unobserved) |
| personal_fouls | integer | Personal fouls in the match, or in the period on period rows; 0 where the era did not rec |
| playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| plus_minus | integer | Point differential while on court in the match, or in the period on period rows; 0 where |
| points | integer | Points in the match, or in the period on period rows; 0 where the era did not record it (n |
| rebounds | integer | Total rebounds in the match, or in the period on period rows; 0 where the era did not re |
| starter | logical | Player started the game. |
| steals | integer | Steals in the match, or in the period on period rows; 0 where the era did not record it (n |
| technical_fouls | integer | Technical fouls in the match, or in the period on period rows; 0 where the era did not re |
| three_points_attempted | integer | Three-point field goals attempted in the match, or in the period on period rows; 0 wher |
| three_points_made | integer | Three-point field goals made in the match, or in the period on period rows; 0 where the |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers | integer | Turnovers in the match, or in the period on period rows; 0 where the era did not record |
| two_points_attempted | integer | Two-point field goals attempted in the match, or in the period on period rows; 0 where |
| two_points_made | integer | Two-point field goals made in the match, or in the period on period rows; 0 where the e |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-by-play row |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the id to jo |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_full_name | character | Given and family name. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to join |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| sc | integer | NBL SuperCoach fantasy score for the game. |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matches(), nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_match_player_box(match_id = "b217a699-4bef-11f0-bc08-933162362355"))
```

---

`nbl_rosetta_match_player_period_box`

# NBL Rosetta Match Player Box Score by Period

---

## Description

One match's player box score by period (Rosetta-computed from the event log), one row per player and period.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}/live/all`

## Usage

```
nbl_rosetta_match_player_period_box(match_id)
```

## Arguments

| | |
|---|---|
| `match_id` | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).

- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_match_player_period_box(match_id = "b217a699-4bef-11f0-bc08-933162362355")
```

## Value

A wehoop_data tibble (one row per player and period):

| col_name | type | description |
|---|---|---|
| match_id | character | DataCore fixture id of the match (UUID v1): the {match_id} route key and fixtureId i |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box score and p |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the id to jo |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take this entity acc |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to joi |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this entity accep |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = first/second/ |
| period_type | character | Period type: regular (overtime unobserved) |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-by-play row |
| assists | integer | Assists in the match, or in the period on period rows; 0 where the era did not record it ( |
| blocks | integer | Blocks in the match, or in the period on period rows; 0 where the era did not record it ( |
| defensive_rebounds | integer | Defensive rebounds in the match, or in the period on period rows; 0 where the era did r |
| field_goals_attempted | integer | Field goals attempted in the match, or in the period on period rows; 0 where the era did |
| field_goals_made | integer | Field goals made in the match, or in the period on period rows; 0 where the era did not |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls | integer | Fouls committed (personal + technical + others) in the match, or in the period on perio |
| free_throws_attempted | integer | Free throws attempted in the match, or in the period on period rows; 0 where the era di |
| free_throws_made | integer | Free throws made in the match, or in the period on period rows; 0 where the era did no |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| minutes | double | Minutes played: "MM:SS" string in player_match_statistics, decimal minutes in calcul |
| offensive_rebounds | integer | Offensive rebounds in the match, or in the period on period rows; 0 where the era did n |
| personal_fouls | integer | Personal fouls in the match, or in the period on period rows; 0 where the era did not re |

| | | |
|---|---|---|
| points | integer | Points in the match, or in the period on period rows; 0 where the era did not record it (r |
| rebounds | integer | Total rebounds in the match, or in the period on period rows; 0 where the era did not re |
| steals | integer | Steals in the match, or in the period on period rows; 0 where the era did not record it (r |
| technical_fouls | integer | Technical fouls in the match, or in the period on period rows; 0 where the era did not re |
| three_points_attempted | integer | Three-point field goals attempted in the match, or in the period on period rows; 0 where |
| three_points_made | integer | Three-point field goals made in the match, or in the period on period rows; 0 where the |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers | integer | Turnovers in the match, or in the period on period rows; 0 where the era did not record |
| two_points_attempted | integer | Two-point field goals attempted in the match, or in the period on period rows; 0 where |
| two_points_made | integer | Two-point field goals made in the match, or in the period on period rows; 0 where the e |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| jersey_number | character | Jersey number, a STRING. |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`, `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

---

`nbl_rosetta_match_team_box`

# NBL Rosetta Match Team Box Score

---

## Description

One match's team box score by period, one row per team and period (0 = whole game).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/match/{match_id}/live/all`

## Usage

```
nbl_rosetta_match_team_box(match_id)
```

## Arguments

| | |
|---|---|
| `match_id` | (*character* required): DataCore fixture id of the match (match_id of nbl_rosetta_matches), a UUID, e.g. b217a699-4bef-11f0-bc08-933162362355. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per team and period):

| col_name | type | description |
|---|---|---|
| match_id | character | DataCore fixture id of the match (UUID v1): the `{match_id}` route key and fixtureId i |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box score and p |
| stat_line_id | character | Provider record hash of this statistics row (64 hex characters) |
| rosetta_stat_line_id | character | Rosetta id of this statistics row (UUID v4) |
| assists | integer | Assists in the match, or in the period on period rows; 0 where the era did not record it ( |
| blocks | integer | Blocks in the match, or in the period on period rows; 0 where the era did not record it ( |
| defensive_rebounds | integer | Defensive rebounds in the match, or in the period on period rows; 0 where the era did r |
| field_goals_attempted | integer | Field goals attempted in the match, or in the period on period rows; 0 where the era dic |
| field_goals_made | integer | Field goals made in the match, or in the period on period rows; 0 where the era did not |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls | integer | Fouls committed (personal + technical + others) in the match, or in the period on perio |
| free_throws_attempted | integer | Free throws attempted in the match, or in the period on period rows; 0 where the era di |
| free_throws_made | integer | Free throws made in the match, or in the period on period rows; 0 where the era did no |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |

| | | |
|---|---|---|
| minutes | double | Minutes played: "MM:SS" string in player_match_statistics, decimal minutes in calcul |
| offensive_rebounds | integer | Offensive rebounds in the match, or in the period on period rows; 0 where the era did n |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = first/second/ |
| period_type | character | Period type: regular (overtime unobserved) |
| personal_fouls | integer | Personal fouls in the match, or in the period on period rows; 0 where the era did not rec |
| points | integer | Points in the match, or in the period on period rows; 0 where the era did not record it (n |
| rebounds | integer | Total rebounds in the match, or in the period on period rows; 0 where the era did not re |
| steals | integer | Steals in the match, or in the period on period rows; 0 where the era did not record it (n |
| technical_fouls | integer | Technical fouls in the match, or in the period on period rows; 0 where the era did not re |
| three_points_attempted | integer | Three-point field goals attempted in the match, or in the period on period rows; 0 where |
| three_points_made | integer | Three-point field goals made in the match, or in the period on period rows; 0 where the |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers | integer | Turnovers in the match, or in the period on period rows; 0 where the era did not record |
| two_points_attempted | integer | Two-point field goals attempted in the match, or in the period on period rows; 0 where |
| two_points_made | integer | Two-point field goals made in the match, or in the period on period rows; 0 where the e |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-by-play row |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to join |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| timeouts | integer | Timeouts taken in the match, or in the period on period rows; 0 where the era did not re |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`, `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

## Examples

```
try(nbl_rosetta_match_team_box(match_id = "b217a699-4bef-11f0-bc08-933162362355"))
```

---

`nbl_rosetta_matches` **NBL Rosetta Matches**

---

## Description

Matches of a season: ids, teams, venue, status, scores and start time (UTC).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/matches/in/season/{season}/{season_type}`

## Usage

```
nbl_rosetta_matches(
  season,
  org = "wnbl",
  season_type = "all",
  limit = NULL,
  offset = NULL
)
```

## Arguments

| | |
|---|---|
| season | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| org | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |
| season_type | (*character* default `"all"`): Season type: all (every type of the year), regular (includes finals), preseason, in_season (cups, Blitz, NBAxNBL, All-Star) or test. One of `"all"`, `"regular"`, `"preseason"`, `"in_season"`, `"test"`. |
| limit | (*integer* default `NULL`): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |
| offset | (*integer* default `NULL`): Rows to skip. Pages can overlap, so prefer one large limit. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.

- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per match):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competit |
| match_id | character | DataCore fixture id (UUID v1); equal to `id`. |
| external_media_id | character | Media id. Through NBL25 (2024-25): the Genius Sports match id (7 digits, |
| date_updated | character | Record last-edit time, ISO 8601 UTC. |
| start_time | character | Tip-off, UTC, with no offset or a Z suffix (ESPN dates the same Perth v SE M |
| start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| round | character | Round label: a number as a string in the regular season ("13"), words in the f |
| match_type | character | Match type: regular, final (FINAL upper case seen) |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box s |
| status | character | Provider fixture status: CONFIRMED, SCHEDULED. |
| home_score | integer | Home final score, a STRING; null before tip-off. |
| away_score | integer | Away final score, a STRING; null before tip-off. |
| attendance | integer | Attendance, a STRING; null when not recorded. |
| match_slug | character | URL slug of the match page on nbl.com.au. |
| match_title | character | Match title ("<home> v <away> <men/women> DD-MM-YYYY") |
| venue_id | character | DataCore id (UUID v1, Sportradar) of the venue; stable across seasons and t |
| rosetta_venue_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the venue; routes that take this |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_suburb | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Country; often null. |
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_code | character | Venue code; often null. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| home_team_id | character | DataCore id (UUID v1, Sportradar) of the home team; stable across seasons |
| rosetta_home_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the home team; routes that take |
| home_team_name | character | Display name of the team. |
| home_team_external_team_name | character | Team name as DataCore spells it. |
| home_team_code | character | Team code of the home team, e.g. PER or SYD. |
| home_team_division_abbreviation | character | Division code, e.g. EM (East Men) |
| home_team_division_name | character | Division name (NBL1 conference + gender) |

| | | |
|---|---|---|
| home_team_conference_name | character | Name of the NBL1 conference of the home team; null outside NBL1. |
| away_team_id | character | DataCore id (UUID v1, Sportradar) of the away team; stable across seasons |
| rosetta_away_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the away team; routes that take |
| away_team_name | character | Display name of the team. |
| away_team_external_team_name | character | Team name as DataCore spells it. |
| away_team_code | character | Team code of the away team, e.g. PER or SYD. |
| away_team_division_abbreviation | character | Division code, e.g. EM (East Men) |
| away_team_division_name | character | Division name (NBL1 conference + gender) |
| away_team_conference_name | character | Name of the NBL1 conference of the away team; null outside NBL1. |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this |
| season_name | character | Season name (NBL26, WNBL 2024/25, 2025 South Men) |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhi |
| season_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female l |
| season_competition_id | character | Rosetta competition id (UUID v4) |
| season_competition_name | character | Name of the competition the season belongs to (e.g. NBL, WNBL, NBL1 N |
| season_competition_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female l |
| season_competition_league_name | character | Name of the league the competition belongs to (NBL, WNBL or NBL1). |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |
| home_team_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| home_team_conference | character | Conference of the division. |
| away_team_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| away_team_conference | character | Conference of the division. |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matches(), nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_matches(org = "nbl", season = 2026, season_type = "regular"))
```

`nbl_rosetta_next_match`

# NBL Rosetta Next Match

## Description

The organisation's next scheduled match (empty between seasons).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/next/match/for/{org}/{season}`

## Usage

```
nbl_rosetta_next_match(season, org = "wnbl")
```

## Arguments

| | |
|---|---|
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.

- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_next_match(org = "nbl", season = 2027)
```

## Value

A `wehoop_data` tibble (one row: the next match):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competit |
| match_id | character | DataCore fixture id (UUID v1); equal to `id`. |
| external_media_id | character | Media id. Through NBL25 (2024-25): the Genius Sports match id (7 digits, e |
| date_updated | character | Record last-edit time, ISO 8601 UTC. |
| start_time | character | Tip-off, UTC, with no offset or a Z suffix (ESPN dates the same Perth v SE M |
| start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| round | character | Round label: a number as a string in the regular season ("13"), words in the f |
| match_type | character | Match type: regular, final (FINAL upper case seen) |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box sc |
| status | character | Provider fixture status: CONFIRMED, SCHEDULED. |
| home_score | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_score | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| attendance | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| match_slug | character | URL slug of the match page on nbl.com.au. |
| match_title | character | Match title ("<home> v <away> <men/women> DD-MM-YYYY") |
| venue_id | character | DataCore id (UUID v1, Sportradar) of the venue; stable across seasons and tl |
| rosetta_venue_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the venue; routes that take this e |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_suburb | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_code | character | Venue code; often null. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| home_team_id | character | DataCore id (UUID v1, Sportradar) of the home team; stable across seasons a |
| rosetta_home_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the home team; routes that take |
| home_team_name | character | Display name of the team. |
| home_team_external_team_name | character | Team name as DataCore spells it. |
| home_team_code | character | Team code of the home team, e.g. PER or SYD. |

| | | |
|---|---|---|
| home_team_division | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| home_team_conference | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_team_id | character | DataCore id (UUID v1, Sportradar) of the away team; stable across seasons a |
| rosetta_away_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the away team; routes that take |
| away_team_name | character | Display name of the team. |
| away_team_external_team_name | character | Team name as DataCore spells it. |
| away_team_code | character | Team code of the away team, e.g. PER or SYD. |
| away_team_division | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_team_conference | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and t |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this |
| season_name | character | Season name (NBL26, WNBL 2024/25, 2025 South Men) |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhi |
| season_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lo |
| season_competition_id | character | Rosetta competition id (UUID v4) |
| season_competition_name | character | Name of the competition the season belongs to (e.g. NBL, WNBL, NBL1 No |
| season_competition_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lo |
| season_competition_league_name | character | Name of the league the competition belongs to (NBL, WNBL or NBL1). |
| season_competition_division | character | Always an empty array in samples; element shape and meaning unverified. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matches()`, `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

---

`nbl_rosetta_next_match_team`

# NBL Rosetta Next Match Team

---

## Description

Next match; the team segment is ignored by Rosetta (the body is the organisation's next match).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/next/match/for/team/{team_code}/{org}/{season}`

## Usage

```
nbl_rosetta_next_match_team(team_code, season, org = "wnbl")
```

## Arguments

| | |
|---|---|
| team_code | (*character* required): Team code (team_code of nbl_rosetta_teams), e.g. PER for the Perth Wildcats. |
| season | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_next_match_team(org = "nbl", season = 2027, team_code = "PER")
```

## Value

A wehoop_data tibble (one row: the next match):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competit |
| match_id | character | DataCore fixture id (UUID v1); equal to `id`. |
| external_media_id | character | Media id. Through NBL25 (2024-25): the Genius Sports match id (7 digits, e |
| date_updated | character | Record last-edit time, ISO 8601 UTC. |
| start_time | character | Tip-off, UTC, with no offset or a Z suffix (ESPN dates the same Perth v SE M |
| start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| round | character | Round label: a number as a string in the regular season ("13"), words in the f |
| match_type | character | Match type: regular, final (FINAL upper case seen) |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box sc |
| status | character | Provider fixture status: CONFIRMED, SCHEDULED. |
| home_score | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_score | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| attendance | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| match_slug | character | URL slug of the match page on nbl.com.au. |
| match_title | character | Match title ("<home> v <away> <men/women> DD-MM-YYYY") |
| venue_id | character | DataCore id (UUID v1, Sportradar) of the venue; stable across seasons and th |
| rosetta_venue_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the venue; routes that take this e |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_suburb | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_code | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| home_team_id | character | DataCore id (UUID v1, Sportradar) of the home team; stable across seasons a |
| rosetta_home_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the home team; routes that take |
| home_team_name | character | Display name of the team. |
| home_team_external_team_name | character | Team name as DataCore spells it. |
| home_team_code | character | Team code of the home team, e.g. PER or SYD. |
| home_team_division | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| home_team_conference | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_team_id | character | DataCore id (UUID v1, Sportradar) of the away team; stable across seasons a |
| rosetta_away_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the away team; routes that take |
| away_team_name | character | Display name of the team. |
| away_team_external_team_name | character | Team name as DataCore spells it. |
| away_team_code | character | Team code of the away team, e.g. PER or SYD. |
| away_team_division | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| away_team_conference | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and t |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this |
| season_name | character | Season name (NBL26, WNBL 2024/25, 2025 South Men) |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhi |
| season_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lo |

| | | |
|---|---|---|
| season_competition_id | character | Rosetta competition id (UUID v4) |
| season_competition_name | character | Name of the competition the season belongs to (e.g. NBL, WNBL, NBL1 No |
| season_competition_gender | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| season_competition_league_name | character | Name of the league the competition belongs to (NBL, WNBL or NBL1). |
| season_competition_division | character | Always an empty array in samples; element shape and meaning unverified. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_matches(), nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leader nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

`nbl_rosetta_next_matches`

# NBL Rosetta Next Matches

---

## Description

Each team's next matches: opponent, venue and start time.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/next/matches/for/teams/in/{org}/{season}`

## Usage

```
nbl_rosetta_next_matches(season, org = "wnbl")
```

## Arguments

| | |
|---|---|
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_next_matches(org = "nbl", season = 2027)
```

## Value

A `wehoop_data` tibble (one row per team and upcoming match):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, else from |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this entity accept it |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| division | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| conference | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| match_slug | character | URL slug of the upcoming match, e.g. sydney-kings-v-nz-breakers-men-14-10-2026. |
| match_title | character | Title of the upcoming match, e.g. Sydney Kings v NZ Breakers men 14-10-2026. |
| start_time_datetime | character | Start of the upcoming match, UTC, ISO 8601 text without an offset. |
| is_home | logical | True when the row's team is the home team of the upcoming match. |
| opponent_name | character | Name of the opponent in the upcoming match. |
| venue | character | Name of the venue (arena) of the upcoming match. |

| | | |
|---|---|---|
| match_id | character | DataCore fixture id of the match (UUID v1): the {match_id} route key and fixtureId in pla |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-by-play rows; no |
| rosetta_opponent_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the opponent; routes that take this entity accep |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from season_year ( |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leader nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

# nbl_rosetta_outcomes **NBL Rosetta Outcomes**

---

## Description

Win-loss record and last 10, 5 and 3 results per team, keyed by team name.

Endpoint: GET https://prod.rosetta.nbl.com.au/get/match/outcomes/for/{org}/teams/in/season/{season}/{

## Usage

```
nbl_rosetta_outcomes(season, org = "wnbl", season_type = "regular")
```

## Arguments

| | |
|---|---|
| season | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |
| season_type | (*character* default "regular"): Season type: all (every type of the year), regular (includes finals), preseason, in_season (cups, Blitz, NBAxNBL, All-Star) or test. One of "all", "regular", "preseason", "in_season", "test". |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_outcomes(org = "nbl", season = 2026, season_type = "regular")
```

## Value

A `wehoop_data` tibble (one row per team name):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, else from the |
| team_name | character | Name of the team, the key of Rosetta's outcomes map (the body carries no team id). |
| wins | integer | Wins of the team in the season type asked for. |
| losses | integer | Losses of the team in the season type asked for. |
| last10 | character | Results of the team's last 10 games as a W/L string, oldest first. |
| last5 | character | Results of the team's last 5 games as a W/L string, oldest first. |
| last3 | character | Results of the team's last 3 games as a W/L string, oldest first. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from season_year (a S |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leader nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

`nbl_rosetta_player_game_log`

# NBL Rosetta Player Game Log

---

## Description

A player's game log for a season: the box-score line of each game (period 0 = whole game) and the game.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/player_boxscores/for/{player_id}/in/season/{s`

## Usage

```
nbl_rosetta_player_game_log(
  player_id,
  season,
  org = "wnbl",
  participated = NULL
)
```

## Arguments

| | |
|---|---|
| `player_id` | (*character* required): Player id: the DataCore person id (player_id of nbl_rosetta_players) or Rosetta's own id; either is accepted. |
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |
| `participated` | (*integer* default `NULL`): 1 = only games the player played in (without it, games on the roster that he sat out too). Sent as `filter[participated]`. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_player_game_log(
 org = "nbl", participated = 1, player_id = "7619d3ea-410d-11f0-9235-2902fc2f936f",
  season = 2026
)
```

## Value

A `wehoop_data` tibble (one row per game):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competi |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = fi |
| playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| points | integer | Points in the game. |
| rebounds | integer | Total rebounds in the game. |
| assists | integer | Assists in the game. |
| blocks | integer | Blocks in the game. |
| steals | integer | Steals in the game. |
| turnovers | integer | Turnovers in the game. |

| | | |
|---|---|---|
| offensive_rebounds | integer | Offensive rebounds in the game. |
| defensive_rebounds | integer | Defensive rebounds in the game. |
| personal_fouls | integer | Personal fouls in the game. |
| three_points_attempted | integer | Three-point field goals attempted in the game. |
| three_points_made | integer | Three-point field goals made in the game. |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| two_points_attempted | integer | Two-point field goals attempted in the game. |
| two_points_made | integer | Two-point field goals made in the game. |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| field_goals_attempted | integer | Field goals attempted in the game. |
| field_goals_made | integer | Field goals made in the game. |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| free_throws_attempted | integer | Free throws attempted in the game. |
| free_throws_made | integer | Free throws made in the game. |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| minutes | double | Minutes played, decimal minutes. |
| participated | logical | Player took part in the game. |
| match_id | character | DataCore fixture id of the match (UUID v1): the `{match_id}` route key and |
| rosetta_match_id | character | Rosetta's match RECORD id (UUID v4) as carried by box-score and play-b |
| match_start_time_datetime | character | Tip-off, UTC; same value as start_time. |
| match_status | character | Status of the match in Rosetta's words (complete for a finished game): box s |
| match_season_id | character | DataCore id (UUID v1, Sportradar) of the match's season; stable across seas |
| rosetta_match_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the match's season; routes that |
| match_season_year | integer | Season year as a STRING. NBL and WNBL from 1998-99: the START yea |
| match_season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exh |
| rosetta_match_home_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the match's home team; routes |
| match_home_team_name | character | Display name of the team. |
| match_home_team_code | character | Team code of the match's home team, e.g. PER or SYD. |
| rosetta_match_away_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the match's away team; routes |
| match_away_team_name | character | Display name of the team. |
| match_away_team_code | character | Team code of the match's away team, e.g. PER or SYD. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take this |
| player_jersey_number | character | Jersey number, a STRING. |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and th |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this e |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| sc_total_score | integer | SuperCoach score including double-double / triple-double bonuses. |
| sc_total_score_without_dd_and_td | integer | SuperCoach score without those bonuses. |
| sc_breakdown_points_value | integer | The stat's value for the game or season. |
| sc_breakdown_points_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_rebounds_value | integer | The stat's value for the game or season. |
| sc_breakdown_rebounds_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_assists_value | integer | The stat's value for the game or season. |
| sc_breakdown_assists_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |

| | | |
|---|---|---|
| sc_breakdown_blocks_value | integer | The stat's value for the game or season. |
| sc_breakdown_blocks_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_steals_value | integer | The stat's value for the game or season. |
| sc_breakdown_steals_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_turnovers_value | integer | The stat's value for the game or season. |
| sc_breakdown_turnovers_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_double_figure_count | integer | Stats in double figures (for the double-double / triple-double bonus) |
| sc_season_total | integer | Player's SuperCoach total for the season so far. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_outcomes(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

`nbl_rosetta_player_stats`

# NBL Rosetta Player Stats

---

## Description

A player's season totals and averages for every season, by period.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/statistics/for/player/{player_id}`

## Usage

```
nbl_rosetta_player_stats(player_id, org = "wnbl")
```

## Arguments

| | |
|---|---|
| `player_id` | (*character* required): Player id: the DataCore person id (player_id of nbl_rosetta_players) or Rosetta's own id; either is accepted. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per season and period):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's comp |
| assists_average | double | Assists per game. |
| blocks_average | double | Blocks per game. |
| field_goals_attempted | integer | Field goals attempted, season total. |
| field_goals_made | integer | Field goals made, season total. |
| field_goals_attempted_average | double | Field goals attempted per game. |
| field_goals_made_average | double | Field goals made per game. |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls_average | double | Fouls committed (personal + technical + others) per game. |
| free_throws_attempted | integer | Free throws attempted, season total. |
| free_throws_made | integer | Free throws made, season total. |
| free_throws_attempted_average | double | Free throws attempted per game. |
| free_throws_made_average | double | Free throws made per game. |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |

| | | |
|---|---|---|
| games | integer | Games played in the season (in the period rows, by the player or team). |
| games_started | integer | Games in the starting five. |
| offensive_rebounds | integer | Offensive rebounds, season total. |
| fouls | integer | Fouls committed (personal + technical + others), season total. |
| personal_fouls | integer | Personal fouls, season total. |
| technical_fouls | integer | Technical fouls, season total. |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = |
| three_points_attempted | integer | Three-point field goals attempted, season total. |
| three_points_attempted_average | double | Three-point field goals attempted per game. |
| three_points_made | integer | Three-point field goals made, season total. |
| three_points_made_average | double | Three-point field goals made per game. |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers_average | double | Turnovers per game. |
| turnovers | integer | Turnovers, season total. |
| two_points_attempted | integer | Two-point field goals attempted, season total. |
| two_points_attempted_average | double | Two-point field goals attempted per game. |
| two_points_made | integer | Two-point field goals made, season total. |
| two_points_made_average | double | Two-point field goals made per game. |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| minutes | double | Minutes played, decimal minutes. |
| minutes_average | double | Minutes per game, decimal minutes. |
| points | integer | Points, season total. |
| rebounds | integer | Total rebounds, season total. |
| assists | integer | Assists, season total. |
| blocks | integer | Blocks, season total. |
| steals | integer | Steals, season total. |
| plus_minus | integer | Point differential while on court, season total. |
| plus_minus_average | double | Point differential while on court per game. |
| efficiency | integer | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - misse |
| efficiency_average | double | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - misse |
| points_average | double | Points per game. |
| rebounds_average | double | Total rebounds per game. |
| steals_average | double | Steals per game. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons a |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take t |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons an |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take th |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons a |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take t |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL fro |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, e |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |

| | | |
|---|---|---|
| sc_total_score | integer | SuperCoach score including double-double / triple-double bonuses. |
| sc_total_score_without_dd_and_td | integer | SuperCoach score without those bonuses. |
| sc_breakdown_points_value | integer | The stat's value for the game or season. |
| sc_breakdown_points_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_rebounds_value | integer | The stat's value for the game or season. |
| sc_breakdown_rebounds_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_assists_value | integer | The stat's value for the game or season. |
| sc_breakdown_assists_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_blocks_value | integer | The stat's value for the game or season. |
| sc_breakdown_blocks_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_steals_value | integer | The stat's value for the game or season. |
| sc_breakdown_steals_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_turnovers_value | integer | The stat's value for the game or season. |
| sc_breakdown_turnovers_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_breakdown_triple_double_value | integer | The stat's value for the game or season. |
| sc_breakdown_triple_double_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| sc_double_figure_count | integer | Stats in double figures (for the double-double / triple-double bonus) |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived fr |
| sc_breakdown_double_double_value | integer | The stat's value for the game or season. |
| sc_breakdown_double_double_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnove |
| season_start_date | character | First day of the season, ISO date-time without offset. |
| player_full_name | character | Given and family name. |
| player_active | logical | Player record is active. |
| player_date_of_birth | character | Date of birth, ISO date. |
| player_nationality | character | Unobserved in samples (null in every capture); type and meaning unverifi |
| player_height | double | Height in centimetres. |
| player_weight | double | Unobserved in samples (null in every capture); type and meaning unverifi |
| team_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/fema |
| synthetic_from_roster | logical | Row built from the roster because the player has no season statistics (all s |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

## Examples

```
try(nbl_rosetta_player_stats(org = "nbl", player_id = "7619d3ea-410d-11f0-9235-2902fc2f936f"))
```

`nbl_rosetta_players` **NBL Rosetta Players**

## Description

Roster rows of every team in a season: player bio, jersey, position and team.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/players/in/season/{season}`

## Usage

```
nbl_rosetta_players(season, org = "wnbl", limit = NULL, offset = NULL)
```

## Arguments

| | |
|---|---|
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |
| `limit` | (*integer* default `NULL`): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |
| `offset` | (*integer* default `NULL`): Rows to skip. Pages can overlap, so prefer one large limit. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).

- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

## Value

A wehoop_data tibble (one row per roster entry):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, e |
| jersey_number | character | Jersey number, a STRING. |
| playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the id |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take this entity |
| player_date_updated | character | Record last-edit time, ISO 8601 UTC. |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_date_of_birth | character | Date of birth, ISO date. |
| player_height | double | Height in centimetres. |
| player_weight | double | Weight in kilograms. |
| player_dominant_hand | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lower |
| player_nationality | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_nationality_code | character | Nationality, ISO 3166-1 alpha-3 code. |
| player_dominant_foot | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| player_active | logical | Player record is active. |
| player_league | character | Rosetta league record id, UUID v4. |
| player_birthplace | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_rookie_year | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_person | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_latest_jersey_number | character | Player's most recent jersey number, a STRING. |
| player_full_name | character | Given and family name. |
| player_source_updated_at | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_source_provider | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id t |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this entity |
| team_code | character | Team code, e.g. PER (the {team_code} path value) |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |

| | | |
|---|---|---|
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from seaso |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leade nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_players(org = "nbl", season = 2026))
```

---

# nbl_rosetta_seasons NBL Rosetta Seasons

---

## Description

Every season of an organisation: regular, preseason, in-season cups and test seasons, with dates.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/seasons`

## Usage

```
nbl_rosetta_seasons(org = "wnbl", limit = NULL)
```

## Arguments

| | |
|---|---|
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |
| limit | (*integer* default NULL): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.

- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per season):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, else |
| season_id | character | DataCore season id (UUID v1); accepted as `{season_id}`. |
| rosetta_season_id | character | Rosetta season id (UUID v4; v5 on NBL1); the `{season_id}` route key. |
| name | character | Season name as Rosetta spells it (NBL26, WNBL 2024/25, 2025 South Men); names |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from 1998-99 |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhibition), |
| start_date | character | First day of the season, ISO date-time without offset. |
| end_date | character | Last day of the season, ISO date-time without offset; null on older seasons. |
| competition_league_name | character | Name of the league the season's competition belongs to (NBL, WNBL or NBL1). |
| competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from season_ |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`,

nbl_rosetta_players(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leader
nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(),
nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_seasons(org = "nbl"))
```

---

nbl_rosetta_seasons_current

# NBL Rosetta Seasons Current

---

## Description

The season in progress (empty between seasons: wehoop_no_data).

Endpoint: GET https://prod.rosetta.nbl.com.au/get/{org}/seasons/current

## Usage

```
nbl_rosetta_seasons_current(org = "wnbl")
```

## Arguments

| | |
|---|---|
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).

- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

## Value

A wehoop_data tibble (one row: the season in progress):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, els |
| season_id | character | DataCore season id (UUID v1); accepted as {season_id}. |
| rosetta_season_id | character | Rosetta season id (UUID v4; v5 on NBL1); the {season_id} route key. |
| name | character | Season name as Rosetta spells it (NBL26, WNBL 2024/25); names drift, key on sea |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from 1998-9 |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhibition), |
| start_date | character | First day of the season, ISO date-time without offset. |
| end_date | character | Last day of the season, ISO date-time without offset; null on older seasons. |
| competition_name | character | Name of the competition the season belongs to (e.g. NBL). |
| competition_age_group | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| next_match_start_datetime | character | Tip-off of the season's next match, UTC; null when none. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from season_ |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_seasons_current(org = "nbl"))
```

`nbl_rosetta_standings` **NBL Rosetta Standings**

## Description

Regular-season ladder: position, record, points for and against, home and away splits and form.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/standings/{season}/regular`

## Usage

```
nbl_rosetta_standings(season, org = "wnbl", limit = NULL)
```

## Arguments

| | |
|---|---|
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |
| `limit` | (*integer* default `NULL`): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.

- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

## Value

A `wehoop_data` tibble (one row per ladder entry):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, e |
| rosetta_standing_id | character | Rosetta's id of this ladder row (UUID v4); not an entity id. |
| position | integer | Ladder position (1 = top) |
| won | integer | Wins in the season. |
| lost | integer | Losses in the season. |
| played | integer | Games played in the season (wins plus losses). |
| points_percentage | double | points_for / points_against, a RATIO (1.1379 = 113.79 %) |
| win_percentage | double | Won / played, a 0-1 FRACTION. |
| last_5 | character | Results string, W/L oldest first. Despite the name it can hold the whole season (33 |
| streak | integer | Current streak length (the type is the last letter of last_5) |
| points_for | integer | Points scored by the team in the season. |
| points_against | integer | Points conceded. |
| away_losses | integer | Away losses in the season. |
| away_wins | integer | Away wins in the season. |
| home_losses | integer | Home losses in the season. |
| home_wins | integer | Home wins in the season. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this entity a |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from 1998- |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhibition |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from seaso |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`,

nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_stats_leaders(
nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(),
nbl_rosetta_teams()

## Examples

```
try(nbl_rosetta_standings(org = "nbl", season = 2026))
```

---

`nbl_rosetta_stats_leaders`

# NBL Rosetta Stats Leaders

---

## Description

Player season totals and averages of a season (the leaderboard), one row per player and period.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/stats/leaders/for/season/id/{season_id}`

## Usage

```
nbl_rosetta_stats_leaders(season_id, org = "wnbl", limit = NULL)
```

## Arguments

| | |
|---|---|
| `season_id` | (*character* required): Season id: the DataCore season id (season_id of nbl_rosetta_seasons) or Rosetta's own id (rosetta_season_id); either is accepted. Not the year. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |
| `limit` | (*integer* default `NULL`): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).

- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_stats_leaders(org = "nbl", season_id = "1f8e4a79-e98b-457b-85a5-e4b898c6c0bd")
```

## Value

A `wehoop_data` tibble (one row per player and period):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's compet |
| assists_average | double | Assists per game. |
| blocks_average | double | Blocks per game. |
| field_goals_attempted | integer | Field goals attempted, season total. |
| field_goals_made | integer | Field goals made, season total. |
| field_goals_attempted_average | double | Field goals attempted per game. |
| field_goals_made_average | double | Field goals made per game. |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls_average | double | Fouls committed (personal + technical + others) per game. |
| free_throws_attempted | integer | Free throws attempted, season total. |
| free_throws_made | integer | Free throws made, season total. |
| free_throws_attempted_average | double | Free throws attempted per game. |
| free_throws_made_average | double | Free throws made per game. |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| games | integer | Games played in the season (in the period rows, by the player or team). |
| games_started | integer | Games in the starting five. |
| offensive_rebounds | integer | Offensive rebounds, season total. |
| fouls | integer | Fouls committed (personal + technical + others), season total. |
| personal_fouls | integer | Personal fouls, season total. |
| technical_fouls | integer | Technical fouls, season total. |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = f |
| three_points_attempted | integer | Three-point field goals attempted, season total. |
| three_points_attempted_average | double | Three-point field goals attempted per game. |
| three_points_made | integer | Three-point field goals made, season total. |
| three_points_made_average | double | Three-point field goals made per game. |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |

| | | |
|---|---|---|
| turnovers_average | double | Turnovers per game. |
| turnovers | integer | Turnovers, season total. |
| two_points_attempted | integer | Two-point field goals attempted, season total. |
| two_points_attempted_average | double | Two-point field goals attempted per game. |
| two_points_made | integer | Two-point field goals made, season total. |
| two_points_made_average | double | Two-point field goals made per game. |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| minutes | double | Minutes played, decimal minutes. |
| minutes_average | double | Minutes per game, decimal minutes. |
| points | integer | Points, season total. |
| rebounds | integer | Total rebounds, season total. |
| assists | integer | Assists, season total. |
| blocks | integer | Blocks, season total. |
| steals | integer | Steals, season total. |
| plus_minus | integer | Point differential while on court, season total. |
| plus_minus_average | double | Point differential while on court per game. |
| efficiency | integer | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - missed |
| efficiency_average | double | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - missed |
| points_average | double | Points per game. |
| rebounds_average | double | Total rebounds per game. |
| steals_average | double | Steals per game. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take thi |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take th |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, ex |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| sc_total_score | integer | SuperCoach score including double-double / triple-double bonuses. |
| sc_total_score_without_dd_and_td | integer | SuperCoach score without those bonuses. |
| sc_breakdown_points_value | integer | The stat's value for the game or season. |
| sc_breakdown_points_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_rebounds_value | integer | The stat's value for the game or season. |
| sc_breakdown_rebounds_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_assists_value | integer | The stat's value for the game or season. |
| sc_breakdown_assists_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_blocks_value | integer | The stat's value for the game or season. |
| sc_breakdown_blocks_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_steals_value | integer | The stat's value for the game or season. |
| sc_breakdown_steals_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |

| | | |
|---|---|---|
| sc_breakdown_turnovers_value | integer | The stat's value for the game or season. |
| sc_breakdown_turnovers_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_breakdown_triple_double_value | integer | The stat's value for the game or season. |
| sc_breakdown_triple_double_points | integer | SuperCoach points this stat earns (e.g. 3 per steal or block, -2 per turnover) |
| sc_double_figure_count | integer | Stats in double figures (for the double-double / triple-double bonus) |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived fro |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_team_players(), nbl_rosetta_team_stats(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

`nbl_rosetta_team_players`

# NBL Rosetta Team Players

---

## Description

Roster of one team in a season: player bio, jersey and position.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/players/for/team/{team_id}/in/season/{season}`

## Usage

```
nbl_rosetta_team_players(team_id, season, org = "wnbl")
```

## Arguments

| | |
|---|---|
| `team_id` | (*character* required): Team id: the DataCore id (team_id of nbl_rosetta_teams) or Rosetta's own id; either is accepted. |
| `season` | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| `org` | (*character* default `"wnbl"`, the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of `"nbl"`, `"wnbl"`, `"nbl1"`. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as `wehoop_asset_fetch`.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: `<entity>`*id columns are DataCore ids (UUID v1; join on them), rosetta*`<entity>`_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (`wehoop_no_data`). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (`wehoop_asset_fetch`); retry after the cache entry expires.

Errors: `wehoop_no_data` when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); `wehoop_asset_fetch` for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; `wehoop_bad_request` for an unknown `org` / `season_type`, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_team_players(
  org = "nbl", season = 2026, team_id = "befee10a-410c-11f0-b3f4-1b3c55488bbb"
)
```

## Value

A `wehoop_data` tibble (one row per roster entry):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, else fr |
| jersey_number | character | Jersey number, a STRING. |
| playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the id to join |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this entity accep |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| team_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lower case |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the id to jo |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take this entity acc |

| | | |
|---|---|---|
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_full_name | character | Given and family name. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| player_active | logical | Player record is active. |
| player_date_of_birth | character | Date of birth, ISO date. |
| player_nationality | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_height | double | Height in centimetres. |
| player_weight | double | Weight in kilograms. |
| player_dominant_hand | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_gender | character | Gender (MALE/FEMALE upper case on seasons and players, male/female lower case |
| player_nationality_code | character | Nationality, ISO 3166-1 alpha-3 code. |
| player_birthplace | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| player_rookie_year | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from season_ye |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`, `nbl_rosetta_teams()`

---

`nbl_rosetta_team_stats`

# NBL Rosetta Team Stats

---

## Description

Team season totals and averages by period (0 = whole game, 1-4 quarters, 5 and up overtime).

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/team/stats/for/season/{season}/regular`

## Usage

```
nbl_rosetta_team_stats(season, org = "wnbl", limit = NULL)
```

## Arguments

| | |
|---|---|
| season | (*integer* required): Season END year (2026 = 2025-26). The runtime maps it to Rosetta's year: the START year for the NBL and WNBL from 1998-99, the calendar year for NBL1 and the NBL / WNBL winter seasons up to 1998. END 1998 and 1999 both read Rosetta year 1998, which holds two seasons; the matches frame keeps the one asked for. |
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |
| limit | (*integer* default NULL): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_team_stats(org = "nbl", season = 2026)
```

## Value

A wehoop_data tibble (one row per team and period):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competitio |
| assists | integer | Assists, team season total (in that period on period rows) |
| assists_average | double | Assists per game. |
| blocks | integer | Blocks, team season total (in that period on period rows) |
| blocks_average | double | Blocks per game. |
| defensive_rebounds | integer | Defensive rebounds, team season total (in that period on period rows) |
| field_goals_attempted | integer | Field goals attempted, team season total (in that period on period rows) |
| field_goals_attempted_average | double | Field goals attempted per game. |
| field_goals_made | integer | Field goals made, team season total (in that period on period rows) |
| field_goals_made_average | double | Field goals made per game. |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls | integer | Fouls committed (personal + technical + others), team season total (in that peri |
| fouls_average | double | Fouls committed (personal + technical + others) per game. |
| free_throws_attempted | integer | Free throws attempted, team season total (in that period on period rows) |
| free_throws_attempted_average | double | Free throws attempted per game. |
| free_throws_made | integer | Free throws made, team season total (in that period on period rows) |
| free_throws_made_average | double | Free throws made per game. |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| games | integer | Games played in the season (in the period rows, by the player or team). |
| minutes | double | Player-minutes played (5 x game minutes) |
| offensive_rebounds | integer | Offensive rebounds, team season total (in that period on period rows) |
| offensive_rebounds_average | double | Offensive rebounds per game. |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = first/ |
| personal_fouls_average | double | Personal fouls per game. |
| points | integer | Points, team season total (in that period on period rows) |
| points_average | double | Points per game. |
| rebounds | integer | Total rebounds, team season total (in that period on period rows) |
| rebounds_average | double | Total rebounds per game. |
| results_string | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| steals | integer | Steals, team season total (in that period on period rows) |
| steals_average | double | Steals per game. |
| three_points_attempted | integer | Three-point field goals attempted, team season total (in that period on period ro |
| three_points_attempted_average | double | Three-point field goals attempted per game. |
| three_points_made | integer | Three-point field goals made, team season total (in that period on period rows) |
| three_points_made_average | double | Three-point field goals made per game. |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers | integer | Turnovers, team season total (in that period on period rows) |
| turnovers_average | double | Turnovers per game. |
| two_points_attempted | integer | Two-point field goals attempted, team season total (in that period on period rov |
| two_points_attempted_average | double | Two-point field goals attempted per game. |
| two_points_made | integer | Two-point field goals made, team season total (in that period on period rows) |
| two_points_made_average | double | Two-point field goals made per game. |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the i |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this enti |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |

| | | |
|---|---|---|
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and the |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this en |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from 19 |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhibi |
| season_competition_division | character | NBL1 divisions (conference x gender); empty for NBL and WNBL. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from se |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: nbl_rosetta_match(), nbl_rosetta_match_by_id(), nbl_rosetta_match_pbp(), nbl_rosetta_match_player_box(), nbl_rosetta_match_player_period_box(), nbl_rosetta_match_team_box(), nbl_rosetta_matches(), nbl_rosetta_next_match(), nbl_rosetta_next_match_team(), nbl_rosetta_next_matche nbl_rosetta_outcomes(), nbl_rosetta_player_game_log(), nbl_rosetta_player_stats(), nbl_rosetta_players(), nbl_rosetta_seasons(), nbl_rosetta_seasons_current(), nbl_rosetta_standings(), nbl_rosetta_stats_leaders(), nbl_rosetta_team_players(), nbl_rosetta_team_stats_leaders(), nbl_rosetta_teams()

---

`nbl_rosetta_team_stats_leaders`

# NBL Rosetta Team Stats Leaders

---

## Description

Player season totals of one team in a season, optionally one period and sorted.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/stats/leaders/for/season/id/{season_id}/team/`

## Usage

```
nbl_rosetta_team_stats_leaders(
  season_id,
  team_code,
  org = "wnbl",
  limit = NULL,
  period = NULL,
  sort = NULL
)
```

## Arguments

| | |
|---|---|
| season_id | (*character* required): Season id: the DataCore season id (season_id of nbl_rosetta_seasons) or Rosetta's own id (rosetta_season_id); either is accepted. Not the year. |
| team_code | (*character* required): Team code (team_code of nbl_rosetta_teams), e.g. PER for the Perth Wildcats. |

| | |
|---|---|
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |
| limit | (*integer* default NULL): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |
| period | (*integer* default NULL): Period to keep: 0 = whole game, 1-4 quarters, 5 and up overtime. Sent as filter[period]. |
| sort | (*character* default NULL): Sort key; a leading minus sorts descending, e.g. -points. |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

```
nbl_rosetta_team_stats_leaders(
  org = "nbl", period = 0, season_id = "1f8e4a79-e98b-457b-85a5-e4b898c6c0bd",
  sort = "-points", team_code = "PER"
)
```

## Value

A `wehoop_data` tibble (one row per player and period):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competitio |
| assists_average | double | Assists per game. |
| blocks_average | double | Blocks per game. |
| field_goals_attempted | integer | Field goals attempted, season total. |
| field_goals_made | integer | Field goals made, season total. |
| field_goals_attempted_average | double | Field goals attempted per game. |
| field_goals_made_average | double | Field goals made per game. |
| field_goals_percentage | double | Field goals made / attempted, a 0-1 FRACTION. |
| fouls_average | double | Fouls committed (personal + technical + others) per game. |
| free_throws_attempted | integer | Free throws attempted, season total. |
| free_throws_made | integer | Free throws made, season total. |
| free_throws_attempted_average | double | Free throws attempted per game. |
| free_throws_made_average | double | Free throws made per game. |
| free_throws_percentage | double | Free throws made / attempted, a 0-1 FRACTION. |
| games | integer | Games played in the season (in the period rows, by the player or team). |
| games_started | integer | Games in the starting five. |
| offensive_rebounds | integer | Offensive rebounds, season total. |
| fouls | integer | Fouls committed (personal + technical + others), season total. |
| personal_fouls | integer | Personal fouls, season total. |
| technical_fouls | integer | Technical fouls, season total. |
| period | integer | Period as a STRING: 0 = game or season total, 1-4 = quarters, 11/12/13 = first/ |
| three_points_attempted | integer | Three-point field goals attempted, season total. |
| three_points_attempted_average | double | Three-point field goals attempted per game. |
| three_points_made | integer | Three-point field goals made, season total. |
| three_points_made_average | double | Three-point field goals made per game. |
| three_points_percentage | double | Three-point field goals made / attempted, a 0-1 FRACTION. |
| turnovers_average | double | Turnovers per game. |
| turnovers | integer | Turnovers, season total. |
| two_points_attempted | integer | Two-point field goals attempted, season total. |
| two_points_attempted_average | double | Two-point field goals attempted per game. |
| two_points_made | integer | Two-point field goals made, season total. |
| two_points_made_average | double | Two-point field goals made per game. |
| two_points_percentage | double | Two-point field goals made / attempted, a 0-1 FRACTION. |
| minutes | double | Minutes played, decimal minutes. |
| minutes_average | double | Minutes per game, decimal minutes. |
| points | integer | Points, season total. |
| rebounds | integer | Total rebounds, season total. |
| assists | integer | Assists, season total. |
| blocks | integer | Blocks, season total. |
| steals | integer | Steals, season total. |
| plus_minus | integer | Point differential while on court, season total. |
| plus_minus_average | double | Point differential while on court per game. |
| efficiency | integer | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - missed FT - |
| efficiency_average | double | Efficiency rating (PTS + REB + AST + STL + BLK - missed FG - missed FT - |

| | | |
|---|---|---|
| points_average | double | Points per game. |
| rebounds_average | double | Total rebounds per game. |
| steals_average | double | Steals per game. |
| player_id | character | DataCore id (UUID v1, Sportradar) of the player; stable across seasons and the |
| rosetta_player_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the player; routes that take this en |
| player_first_name | character | The player's given name, e.g. Elijah. |
| player_jersey_number | character | Jersey number, a STRING. |
| player_last_name | character | The player's family name, e.g. Pepper. |
| player_playing_position | character | Playing position code (G, F, C, PG, SF, FC, ...); null when not recorded. |
| team_id | character | DataCore id (UUID v1, Sportradar) of the team; stable across seasons and the i |
| rosetta_team_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the team; routes that take this enti |
| team_name | character | Name of the team as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the `{team_code}` path value) |
| season_id | character | DataCore id (UUID v1, Sportradar) of the season; stable across seasons and the |
| rosetta_season_id | character | Rosetta's own id (UUID v4; v5 on NBL1) of the season; routes that take this en |
| season_year | integer | Rosetta's year of the season: the START year for the NBL and WNBL from 19 |
| season_type | character | Season type: regular (includes finals), preseason, in_season (cups, blitz, exhibi |
| season_competition_division | character | Always an empty array in samples; element shape and meaning unverified. |
| season_end_year | integer | sdv season key: the END year of the season (2026 = 2025-26), derived from se |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_teams()`

---

`nbl_rosetta_teams` **NBL Rosetta Teams**

---

## Description

Every team record of an organisation: current, defunct, exhibition and development sides.

Endpoint: `GET https://prod.rosetta.nbl.com.au/get/{org}/teams`

## Usage

```
nbl_rosetta_teams(org = "wnbl", limit = NULL)
```

## Arguments

| | |
|---|---|
| org | (*character* default "wnbl", the women's WNBL): Organisation: nbl (men's NBL), wnbl (women's) or nbl1 (the second tier, men's and women's conferences). One of "nbl", "wnbl", "nbl1". |
| limit | (*integer* default NULL): Rows to return. Omitted = a page large enough for the whole list (a short page cut by Rosetta's path cache is refused). |

## Details

- Unofficial: the JSON proxy behind nbl.com.au (not an API the NBL documents or supports). The terms of use were not reviewed; payloads are not redistributed as release assets.
- Gated by a referer, not a key: every request carries Origin / Referer set to nbl.com.au (the runtime sends them); without them the host answers 403 MISSING_API_KEY, which is raised as wehoop_asset_fetch.
- Fills a hole in ESPN: ESPN's nbl feed has box scores but no plays for the 2025-26 NBL season (the stats provider changed from Genius Sports to Sportradar DataCore that year). nbl_rosetta_match_pbp has the full event log with shot x/y for 2025-26 and, in the same shape, back to 2011-12 (WNBL about 2014-15). Matches from 1979 have box scores only.
- One host serves three organisations; pass org='nbl' (default), 'wnbl' or 'nbl1'. NBL1 is the second tier (men's and women's conferences, calendar-year seasons).
- Season = the END year (2026 = 2025-26) everywhere. Rosetta's own year is the START year for the NBL and WNBL from 1998-99 and a calendar year for NBL1 and the 1979-1997 winter seasons; the runtime maps it. Year 1998 holds two seasons (calendar 1998 and 1998-99).
- Ids: <entity>*id columns are DataCore ids (UUID v1; join on them), rosetta*<entity>_id Rosetta's own. match_id is the DataCore fixture id every match route is keyed on.
- Unknown ids, orgs and types answer 200 with no rows (wehoop_no_data). Rosetta caches a body per path and serves it whatever the query string, so a page cut by someone else's limit is refused (wehoop_asset_fetch); retry after the cache entry expires.

Errors: wehoop_no_data when Rosetta answers 404 or no rows (an unknown match, team, player or season id, an org or season type with nothing, or no season in progress); wehoop_asset_fetch for the 403 gate body, a stub or error envelope, a page cut by Rosetta's path cache, a body for another org, season or entity than the one asked for, or any other failed fetch; wehoop_bad_request for an unknown org / season_type, a malformed id, or HTTP 400 / 422.

## Value

A wehoop_data tibble (one row per team record):

| col_name | type | description |
|---|---|---|
| org | character | Organisation the row belongs to: nbl, wnbl or nbl1. From the row's competition, |
| team_id | character | DataCore team id (UUID v1); stable across seasons. |
| rosetta_team_id | character | Rosetta team id (UUID v4; v5 on NBL1); stable across seasons; accepted as {te |
| name | character | Team name as Rosetta spells it, e.g. Perth Wildcats. |
| team_code | character | Team code, e.g. PER (the {team_code} path value) |
| established | character | Year the club was founded; often null. |

| | | |
|---|---|---|
| head_coach | character | Head coach; often null. |
| championships | character | Championships won; often null. |
| website | character | Club website URL. |
| external_media_competitor_id | character | Team id on the media/OTT service; often null. |
| conference | character | NBL1 conference of the team (name); null for NBL and WNBL teams. |
| division | character | NBL1 division of the team (name); null for NBL and WNBL teams. |
| venue_name | character | Name of the venue (arena), e.g. RAC Arena. |
| venue_address | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_state | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_country | character | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_timezone | character | IANA time zone of the venue, e.g. Australia/Sydney. |
| venue_latitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_longitude | double | Unobserved in samples (null in every capture); type and meaning unverified. |
| venue_capacity | integer | Unobserved in samples (null in every capture); type and meaning unverified. |
| conference_name | character | Name of the NBL1 conference the team plays in (e.g. North, South); null outside |
| division_name | character | Division name (NBL1 conference + gender) |

Columns the body adds beyond these follow, in body order (sdv-py keeps them too).

## See Also

Other NBL Rosetta Functions: `nbl_rosetta_match()`, `nbl_rosetta_match_by_id()`, `nbl_rosetta_match_pbp()`, `nbl_rosetta_match_player_box()`, `nbl_rosetta_match_player_period_box()`, `nbl_rosetta_match_team_box()`, `nbl_rosetta_matches()`, `nbl_rosetta_next_match()`, `nbl_rosetta_next_match_team()`, `nbl_rosetta_next_matche` `nbl_rosetta_outcomes()`, `nbl_rosetta_player_game_log()`, `nbl_rosetta_player_stats()`, `nbl_rosetta_players()`, `nbl_rosetta_seasons()`, `nbl_rosetta_seasons_current()`, `nbl_rosetta_standings()`, `nbl_rosetta_stats_leaders()`, `nbl_rosetta_team_players()`, `nbl_rosetta_team_stats()`, `nbl_rosetta_team_stats_leaders()`

## Examples

```
try(nbl_rosetta_teams(org = "nbl"))
```

---

`nbl1_competition` **NBL1 (Australia) (Atrium shortcuts)**

---

## Description

**Per-competition shortcuts over the Atrium embed API for NBL1 (Australia) (`website_id = 3`; men's and women's).**

Thin bindings of the parameterised `atrium_*()` wrappers to this site, minted from the competition registry (sdv-py `tools/codegen/basketball_competitions.yaml`); the full surface stays on the `atrium_*()` wrappers.

## Usage

```
nbl1_seasons(locale = NULL)
nbl1_schedule(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
nbl1_standings(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
nbl1_teams(season = NULL, season_id = NULL, competition_id = NULL, locale = NULL)
nbl1_player_box(fixture_id, locale = NULL)
nbl1_team_box(fixture_id, locale = NULL)
nbl1_pbp(fixture_id, locale = NULL)
nbl1_shots(fixture_id, locale = NULL)
```

## Arguments

| | |
|---|---|
| season | (*integer* optional): Season END year (season rule below). Omit season and season_id for the site's current season. |
| season_id | (*character* optional): Season UUID (season_id in the seasons listing); skips the END-year lookup. |
| competition_id | (*character* optional): Competition UUID (a site carries many: conferences, cups); needed when several share the season. |
| locale | (*character* optional): Label locale, e.g. "en-EN"; omitted = the site's own language. |
| fixture_id | (*character* required): Fixture UUID (fixture_id in the schedule). |

## Details

Season rule: calendar year, Mar-Aug (END year); conferences are competition_ids.

Errors and terms of use are those of atrium_fixtures(): unofficial use of the public widget endpoint, at least 1 s between requests, not republished as release assets. The site also serves women's competitions (pass their competition_id).

```
nbl1_seasons()
nbl1_schedule()
```

## Value

A wehoop_data tibble with the columns of the base wrapper, one table per function:

nbl1_seasons():

The columns of atrium_seasons():

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, |
| season_id | character | Season UUID; pass it as season_id to any season-scoped wrapper. |
| competition_id | character | Competition UUID; pass it as competition_id to pick one of several seasons of an END year. |
| competition_name | character | Competition name in the site's language; the same name can appear for several years. |
| season_name | character | Season name in the site's language, e.g. Betclic ELITE 2025. |
| abbreviation_local | character | Local abbreviation of the season; null in every body seen. |
| year | integer | The site's own year of the season: the START year on LNB (site 12), the calendar year elsew |

| | | |
|---|---|---|
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year on the |
| is_default_season | logical | True for the season the body is about (the site's default when none was asked). |

`nbl1_schedule()`:

The columns of `atrium_fixtures()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| season_id | character | Season UUID the page is about (the site's default season when none was asked). |
| competition_id | character | Competition UUID of the season. |
| competition_name | character | Competition name in the site's language, e.g. Betclic ELITE or LIGA CALIENTE.MX |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| fixture_type | character | Kind of game as sent: REGULAR, PLAYOFF, ... (the set is open). |
| name | character | Fixture name as the site prints it (e.g. Paris - Lyon-Villeurbanne); null on LNBP. |
| round | character | Round of the game as the site words it, e.g. Journee: 30. |
| round_code | character | Round code; null in every body seen. |
| pool | character | Pool the game belongs to; null when the competition has none. |
| series | character | Playoff series the game belongs to as sent (JSON text when an object); null outside a s |
| stage_code | character | Opaque 30-hex stage code of the game; null in every body seen. |
| status | character | Status code of the game, e.g. CONFIRMED (final), as sent. |
| status_label | character | Status of the game as the site words it in its own language (e.g. Termine, Final). |
| start_time_utc | POSIXct | Scheduled start, naive datetime in UTC (the API sends no offset). |
| start_time_local | character | Scheduled start in the venue's local time, as the API sends it (YYYY-MM-DDTHH:MM:SS, |
| timezone | character | IANA time zone of the venue (e.g. Europe/Paris, Mexico/General) that start_time_loca |
| times_unconfirmed | logical | True while the start time is not final. |
| venue | character | Name of the arena where the game is played. |
| venue_id | character | Venue UUID; joins no other route. |
| attendance | integer | Spectators at the game; null when not reported. |
| home_entity_id | character | Team UUID of the home team. |
| home_code | character | Short code of the home team (e.g. PAB). |
| home_name | character | Name of the home team. |
| home_score | integer | Points scored by the home team (the API sends a string; Int64 here). |
| home_result_place | integer | Result place of the home team: 1 for the winner, 2 for the loser. |
| home_result_status | character | Result status of the home team's score, e.g. CONFIRMED. |
| home_draw | logical | True when the game ended level for the home team. |
| home_shootout_attempts | character | Shootout attempts of the home team as sent (an empty string becomes null). |
| home_secondary_score | character | Secondary score of the home team (unused in the samples; shown as text). |
| away_entity_id | character | Team UUID of the away team. |
| away_code | character | Short code of the away team (e.g. PAB). |
| away_name | character | Name of the away team. |
| away_score | integer | Points scored by the away team (the API sends a string; Int64 here). |
| away_result_place | integer | Result place of the away team: 1 for the winner, 2 for the loser. |
| away_result_status | character | Result status of the away team's score, e.g. CONFIRMED. |
| away_draw | logical | True when the game ended level for the away team. |
| away_shootout_attempts | character | Shootout attempts of the away team as sent (an empty string becomes null). |
| away_secondary_score | character | Secondary score of the away team (unused in the samples; shown as text). |

`nbl1_standings()`:
The columns of `atrium_standings()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pana |
| season_id | character | Season UUID of the table. |
| season_end_year | integer | Season END year (the sdv key): the site's year + 1 on LNB (site 12), the calendar year c |
| group | character | Label of the table when the standings hold several groups; null for a single table. |
| position | integer | League table position (1 = first). |
| entity_id | character | Team UUID of the row. |
| team_name | character | Team name as the site prints it. |
| is_current_team | logical | True for the team the page was opened for (team context); false on a plain standings req |
| logo_url | character | URL of the team logo image. |
| wins | integer | Games won; -1 for two teams in the 2025-26 LNB table (an administrative ranking, unv |
| losses | integer | Games lost (team, this table). |
| played | integer | Games played (LNBP and the compact standings) (team, this table). |
| standing_points | integer | League table points (LNBP) (team, this table). |
| win_percentage | double | Share of games won as a 0-1 fraction (not 0-100) (team, this table). |
| scored_for | integer | Points scored over the season (team, this table). |
| scored_against | integer | Points conceded over the season (team, this table). |
| point_difference | integer | Point differential over the season (team, this table). |
| result_string_home | character | Home results as a string of result letters in the site's language, one per game (order and l |
| scored_for_home | integer | Points scored in home games (team, this table). |
| scored_against_home | integer | Points conceded in home games (team, this table). |
| point_difference_home | integer | Point differential in home games (team, this table). |
| result_string_away | character | Away results as a string of result letters in the site's language, one per game (order and l |
| scored_for_away | integer | Points scored in away games (team, this table). |
| scored_against_away | integer | Points conceded in away games (team, this table). |
| point_difference_away | integer | Point differential in away games (team, this table). |
| streak | integer | Length of the current run of results (the win / loss sign convention is unverified) (team, t |
| streak_home | integer | Length of the current run of home results (sign convention unverified) (team, this table). |
| streak_away | integer | Length of the current run of away results (sign convention unverified) (team, this table). |

`nbl1_teams()`:
The columns of `atrium_teams()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 236 Au |
| season_id | character | Season UUID of the team list. |
| entity_id | character | Team UUID (entity): one id per club across seasons, shared by every site that shows the team. |
| name | character | Team name in the site's language. |
| image_url | character | URL of the Sportradar-hosted image (400 px). |

`nbl1_player_box()`:
The columns of `atrium_box_score()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Pa |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID the player is rostered with for this game. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID (person), shared across seasons and sites. |
| person_name | character | Player name as the site prints it (upper case on LNBP and NBL1). |
| bib | character | Jersey number the player wears, as text. |
| position | character | Position code on the roster (LNB, e.g. PG); null on LNBP. |
| starter | logical | True when the player started the game. |
| active | logical | True when the player is on the court in a live game. |
| participated | logical | True when the player entered the game; false rows have null statistics. |
| did_not_play_reason | character | Reason the player did not play; null in every body seen. |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its players |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (player, this game). |
| points | integer | Points scored (player, this game). |
| rebounds | integer | Total rebounds (player, this game). |
| rebounds_offensive | integer | Offensive rebounds (player, this game). |
| rebounds_defensive | integer | Defensive rebounds (player, this game). |
| assists | integer | Assists (player, this game). |
| steals | integer | Steals (player, this game). |
| blocks | integer | Shots blocked (player, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (player, this game). |
| turnovers | integer | Turnovers (player, this game). |
| fouls_total | integer | Fouls committed (player, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (player, this game). |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lines) ( |
| efficiency | integer | Efficiency rating, from the site's own formula (player, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (player, this game). |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (player, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (player, this game). |
| points_two_made | integer | Two-point field goals made (player, this game). |
| points_two_attempted | integer | Two-point field goals attempted (player, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (player, this game). |
| points_three_made | integer | Three-point field goals made (player, this game). |
| points_three_attempted | integer | Three-point field goals attempted (player, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (player, this game). |
| free_throws_made | integer | Free throws made (player, this game). |
| free_throws_attempted | integer | Free throws attempted (player, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (player, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (player, this game). |

`nbl1_team_box()`:

The columns of `atrium_team_box_score()`:

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 2 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail route |
| season_id | character | Season UUID of the game. |
| entity_id | character | Team UUID (from the fixture's isHome flag). |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome fla |
| minutes | character | Time played as an ISO 8601 duration (PT7M23S); a team's is the sum over its pla |
| seconds_played | integer | Time played in whole seconds, parsed from minutes (team, this game). |
| points | integer | Points scored (team, this game). |
| rebounds | integer | Total rebounds (team, this game). |
| rebounds_offensive | integer | Offensive rebounds (team, this game). |
| rebounds_defensive | integer | Defensive rebounds (team, this game). |
| assists | integer | Assists (team, this game). |
| steals | integer | Steals (team, this game). |
| blocks | integer | Shots blocked (team, this game). |
| blocks_received | integer | Own shots blocked by an opponent (LNB only) (team, this game). |
| turnovers | integer | Turnovers (team, this game). |
| fouls_total | integer | Fouls committed (team, this game). |
| fouls_drawn | integer | Fouls drawn (committed against the player or team; LNBP only) (team, this game |
| plus_minus | integer | Plus/minus, the point differential while on court (LNB only; null in some team lin |
| efficiency | integer | Efficiency rating, from the site's own formula (team, this game). |
| field_goals_made | integer | Field goals made, two- and three-pointers together (LNBP only) (team, this game |
| field_goals_attempted | integer | Field goals attempted (LNBP only) (team, this game). |
| field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, this game). |
| points_two_made | integer | Two-point field goals made (team, this game). |
| points_two_attempted | integer | Two-point field goals attempted (team, this game). |
| points_two_percentage | double | Two-point percentage, 0-100 (team, this game). |
| points_three_made | integer | Three-point field goals made (team, this game). |
| points_three_attempted | integer | Three-point field goals attempted (team, this game). |
| points_three_percentage | double | Three-point percentage, 0-100 (team, this game). |
| free_throws_made | integer | Free throws made (team, this game). |
| free_throws_attempted | integer | Free throws attempted (team, this game). |
| free_throws_percentage | double | Free-throw percentage, 0-100 (team, this game). |
| points_fast_break | integer | Fast-break points (LNBP only) (team, this game). |
| extra_fouls_total | integer | Fouls committed (team, credited to the team rather than a player). |
| extra_rebounds | integer | Total rebounds (team, credited to the team rather than a player). |
| extra_rebounds_defensive | integer | Defensive rebounds (team, credited to the team rather than a player). |
| extra_rebounds_offensive | integer | Offensive rebounds (team, credited to the team rather than a player). |
| extra_turnovers | integer | Turnovers (team, credited to the team rather than a player). |
| total_biggest_lead | integer | Biggest lead of the game, in points (team, game total). |
| total_biggest_scoring_run | integer | Longest unanswered scoring run of the game, in points (team, game total). |
| total_field_goals_percentage | double | Field-goal percentage, 0-100 (LNBP only) (team, game total). |
| total_free_throws_percentage | double | Free-throw percentage, 0-100 (team, game total). |
| total_lead_changes | integer | Number of lead changes in the game (team, game total). |
| total_points_fast_break | integer | Fast-break points (LNBP only) (team, game total). |
| total_points_from_bench | integer | Points scored by non-starters (LNBP only) (team, game total). |
| total_points_from_turnover | integer | Points scored off opponent turnovers (LNBP only) (team, game total). |
| total_time_in_lead | character | Time the team led, as an ISO 8601 duration (PT36M11S) (team, game total). |
| total_time_in_lead_seconds | integer | Time the team led, in whole seconds (team, game total). |

| | | |
|---|---|---|
| total_turnovers | integer | Turnovers (team, game total). |

`nbl1_pbp()`:

The columns of `atrium_pbp()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the event in the game (periods in order, events chronological). |
| event_id | character | Event UUID, shared with the same event in atrium_shots. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID; null for team events (time-outs, team rebounds). |
| person_name | character | Player the event is credited to; null for team events (time-outs, team rebounds). |
| bib | character | Jersey number of the player, as text; null for team events. |
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when a shot or free throw was made, false when missed; null for other events. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

`nbl1_shots()`:

The columns of `atrium_shots()`:

| col_name | type | description |
|---|---|---|
| website_id | integer | Site config id the body came from (12 LNB, 14 LNBP, 3 NBL1, 13 El Salvador, 22 Panama, 23 |
| fixture_id | character | Game UUID (fixture); the key of every game frame and of the fixture_detail routes. |
| season_id | character | Season UUID of the game. |
| sequence | integer | 1-based position of the shot among the shots of the game (not the pbp sequence). |
| event_id | character | Event UUID of the shot; the same event in atrium_pbp. |
| period_id | integer | Period number: 1-4 regulation, 11, 12, ... overtimes (not 5). |
| clock | character | Time REMAINING in the period as an ISO 8601 duration (PT10M0S at the start). |
| clock_seconds | integer | Seconds remaining in the period, parsed from clock. |
| entity_id | character | Team UUID the event is credited to. |
| side | character | Side of the game the row belongs to: home or away (from the fixture's isHome flag). |
| person_id | character | Player UUID of the shooter. |
| person_name | character | Name of the shooter. |
| bib | character | Jersey number of the player, as text; null for team events. |

| | | |
|---|---|---|
| event_type | character | Event type: 2pt, 3pt, freeThrow, rebound, assist, steal, block, turnover, foul, substitution, timeO |
| event_sub_type | character | Event detail: jumpShot, layup, offensive, defensive, in, out, 1Of2, personal, ...; null when none. |
| description | character | The event's text in the site's language (e.g. Entre-deux remporte). |
| success | logical | True when the shot was made, false when missed. |
| success_label | character | The made / missed label in the site's language; sent on shot events only. |
| x | double | Shot location along the 28 m court length as percent of the court (0-100, origin at a corner); the |
| y | double | Shot location across the 15 m width as percent of the court (0-100); which sideline is 0 is unveri |
| home_score | integer | Home team's running score after the event. |
| away_score | integer | Away team's running score after the event. |

## See Also

Other Atrium Functions: `atrium_box_score()`, `atrium_competition_player_stats()`, `atrium_competition_team_st`
`atrium_end_year()`, `atrium_fixture()`, `atrium_fixture_periods()`, `atrium_fixtures()`, `atrium_fixtures_ribbon`
`atrium_leaders()`, `atrium_pbp()`, `atrium_player()`, `atrium_player_game_log()`, `atrium_player_seasons()`,
`atrium_player_stats()`, `atrium_players()`, `atrium_season_id()`, `atrium_seasons()`, `atrium_shots()`,
`atrium_standings()`, `atrium_standings_summary()`, `atrium_team()`, `atrium_team_box_score()`,
`atrium_team_stats()`, `atrium_teams()`, `lmb_competition`, `lnbp_competition`, `lpb_competition`

---

`ncaa_wbb` **NCAA Women's Basketball Endpoint Overview**

---

## Description

Wrappers around NCAA-side women's college basketball endpoints beyond what ESPN exposes (rankings, teams reference, etc.). All requests use the same `.retry_request()` HTTP layer as the rest of the package.

## Details

### NCAA endpoints:

| Function | Purpose |
|---|---|
| `ncaa_wbb_teams()` | NCAA WBB team reference |
| `ncaa_wbb_NET_rankings()` | NCAA NET rankings |

### Resilience pattern:

NCAA endpoints occasionally drop columns. The wrappers use `dplyr::any_of()` / `dplyr::rename(any_of(c(new = "old")))` so schema drift doesn't crash the parser — see `ncaa_wbb_NET_rankings()` for the reference implementation.

## ncaa_wbb_NET_rankings *Get Women's college basketball NET rankings for the current date from the NCAA website*

### Description

Get Women's college basketball NET rankings for the current date from the NCAA website

### Usage

```
ncaa_wbb_NET_rankings()
```

### Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### Examples

```
# Get current NCAA NET rankings

  try(ncaa_wbb_NET_rankings())
```

## ncaa_wbb_teams **Scrape NCAA Women's Basketball Teams (Division I, II, and III)**

### Description

Scrapes the NCAA Stats per-season team list at `http://stats.ncaa.org/team/inst_team_list?academic_year=YYYY&c`
Returns one row per team for the requested season-year and division, with the per-season `season_id` and `team_id` parsed out of the `stats.ncaa.org` team URL.

**Network access:** as of v3.0.0 `stats.ncaa.org` is fronted by Akamai and returns HTTP 403 (`Reference #18.<...>`) to many residential and cloud IP ranges regardless of headers. If you hit that block, supply a proxy that egresses from a non-blocked address — pass `proxy =` directly via `...` (forwarded to `.retry_request()`), or set it once with `options(wehoop.proxy = list(url=, port=, username=, password=))`. See the wehoop CLAUDE.md / pkgdown docs for the full proxy resolution order.

### Usage

```
ncaa_wbb_teams(year = most_recent_wbb_season(), division = 1, ...)
```

### Arguments

| | |
|---|---|
| year | The season for which data should be returned, in the form of "YYYY". Years currently available: 2002 onward. |
| division | Division - 1, 2, or 3. |
| ... | Forwarded to the internal .retry_request() HTTP helper. The most useful pass-through is proxy = (string "http://host:port" or named list list(url=, port=, username=, password=, auth=)); see ?.retry_request for the full set of recognized arguments. |

### Details

```
ncaa_wbb_teams(year = 2025, division = 1)

# Behind a proxy (per-call):
ncaa_wbb_teams(
  year = 2025, division = 1,
  proxy = list(url = "http://my-proxy", port = 8080,
               username = "user", password = "pass")
)

# Or session-wide:
options(wehoop.proxy = list(url = "http://my-proxy", port = 8080,
                            username = "user", password = "pass"))
ncaa_wbb_teams(year = 2025, division = 1)
```

### Value

A data frame with the following variables

See the HTML help or pkgdown reference for the column table.

---

## parameter_descriptions

**WNBA Stats API Parameter Descriptions** *Reference table for parameters used across* wnba_*() *functions.*

---

### Description

**WNBA Stats API Parameter Descriptions** Reference table for parameters used across wnba_*() functions.

### Usage

```
parameter_descriptions
```

### Format

A data frame with 52 rows and 5 variables:

| col_name | description |
|---|---|
| parameter | character. R-side argument name (snake_case). |
| api_name | character. WNBA Stats API CamelCase parameter name. |
| description | character. One-sentence description of the parameter. |
| valid_values | character. Comma-separated enum values; empty string for free-form parameters. |
| default | character. Default value as it appears in function signatures. |

---

## parquet_from_url — Load .parquet file from a remote connection

### Description

Sibling of `rds_from_url()` for release assets published as parquet (e.g. the `wnba_stats_leaguedash` / `nba_stats_leaguedash` cube tags). 404-safe: an unpublished asset returns an empty `data.table` + a warning, matching `rds_from_url()`'s contract, instead of raising.

### Usage

```
parquet_from_url(url)
```

### Arguments

| | |
|---|---|
| url | a character url |

### Value

a dataframe as created by `arrow::read_parquet()`

---

## progressively — *Progressively*

### Description

This function helps add progress-reporting to any function - given function `f()` and progressor `p()`, it will return a new function that calls `f()` and then (on-exiting) will call `p()` after every iteration.

### Usage

```
progressively(f, p = NULL)
```

### Arguments

| | |
|---|---|
| f | a function to add progressr functionality to. |
| p | a progressor function as created by progressr::progressor() |

### Details

This is inspired by purrr's safely, quietly, and possibly function decorators.

### Value

a function that does the same as f but it calls p() after iteration.

---

## rds_from_url — Load .rds file from a remote connection

---

### Description

**Load .rds file from a remote connection**

### Usage

```
rds_from_url(url)
```

### Arguments

| | |
|---|---|
| url | a character url |

### Value

a dataframe as created by readRDS()

---

## scb_game — Get BBL / DBBL Game

---

### Description

One game header: teams, score by period, officials, venue and top scorer.

Endpoint: GET https://api.basketball-bundesliga.de/games/{game_id} (DBBL: the same route on https://api-1.dbbl.scb.world).

### Usage

```
scb_game(game_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| game_id | (required) Game id (game_id in scb_games), e.g. 2006920 (BBL) or 10130 (DBBL). One sequence per league. |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: wehoop_no_data (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), wehoop_bad_request (a bad argument, another 400 / 422) and wehoop_asset_fetch (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting wehoop_error. Wrappers raise; they never return an empty frame.

## Value

A wehoop_data tibble: One game header: teams, score by period, officials, venue and top scorer. An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_game):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBBL game ic |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's season_id plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| game_id | character | Game id, a string; unique within a league only (BBL and DBBL overlap), and not |

| | | |
|---|---|---|
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CUP on the E |
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, FIRST_ROUNI |
| status | character | Game status: PRE (scheduled), LIVE, POST (ended, not yet official), OFFICIAL |
| stat_status | character | Live-stats feed status code. Observed: E (ended; German Ende), N (not started), 1 |
| game_progress | character | Live progress code: E (ended), empty or null before the game. Meaning inferred. |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |
| match_day | integer | Round number within the stage (Spieltag). |
| is_tbd | logical | Date or time not yet fixed. |
| is_hidden | logical | True when the API hides the game (0/1 on /games, a boolean on the header). |
| home_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| home_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| home_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| home_tlc | character | Three-letter club code. |
| home_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| home_rank | integer | The team's current table position. |
| home_streak | character | Current streak: W or L; empty when none. |
| home_streak_count | integer | Length of the current streak. |
| home_last10_victories | integer | Wins in the last 10 games. |
| home_last10_losses | integer | Losses in the last 10 games. |
| home_total_games | integer | Games the home team has played so far (the API sends it as a string). |
| guest_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| guest_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| guest_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| guest_tlc | character | Three-letter club code. |
| guest_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| guest_rank | integer | The team's current table position. |
| guest_streak | character | Current streak: W or L; empty when none. |
| guest_streak_count | integer | Length of the current streak. |
| guest_last10_victories | integer | Wins in the last 10 games. |
| guest_last10_losses | integer | Losses in the last 10 games. |
| guest_total_games | integer | Games the guest team has played so far (the API sends it as a string). |
| home_score | integer | Home team final score. |
| guest_score | integer | Guest team final score. |
| latest_period | integer | Number of the last period played (4 = Q4, 5 = OT1, ...); the API sends `Q4` on /gam |
| spectators | integer | Attendance; 0 when not recorded. |
| home_q1_score | integer | Home team points in quarter 1; 0 when quarter scores were not recorded. |
| home_q2_score | integer | Home team points in quarter 2; 0 when quarter scores were not recorded. |
| home_q3_score | integer | Home team points in quarter 3; 0 when quarter scores were not recorded. |
| home_q4_score | integer | Home team points in quarter 4; 0 when quarter scores were not recorded. |
| home_ot1_score | integer | Home team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| home_ot2_score | integer | Home team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| home_ot3_score | integer | Home team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| home_ot4_score | integer | Home team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| home_ot5_score | integer | Home team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| home_ot6_score | integer | Home team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| home_ot7_score | integer | Home team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| home_ot8_score | integer | Home team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| home_ot9_score | integer | Home team points in overtime 9; 0 when not played (null from OT5 on in old seas |

| | | |
|---|---|---|
| guest_q1_score | integer | Guest team points in quarter 1; 0 when quarter scores were not recorded. |
| guest_q2_score | integer | Guest team points in quarter 2; 0 when quarter scores were not recorded. |
| guest_q3_score | integer | Guest team points in quarter 3; 0 when quarter scores were not recorded. |
| guest_q4_score | integer | Guest team points in quarter 4; 0 when quarter scores were not recorded. |
| guest_ot1_score | integer | Guest team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| guest_ot2_score | integer | Guest team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| guest_ot3_score | integer | Guest team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| guest_ot4_score | integer | Guest team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| guest_ot5_score | integer | Guest team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| guest_ot6_score | integer | Guest team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| guest_ot7_score | integer | Guest team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| guest_ot8_score | integer | Guest team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| guest_ot9_score | integer | Guest team points in overtime 9; 0 when not played (null from OT5 on in old seas |
| referee1_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee1_first_name | character | Official's given name. |
| referee1_last_name | character | Official's family name. |
| referee2_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee2_first_name | character | Official's given name. |
| referee2_last_name | character | Official's family name. |
| referee3_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee3_first_name | character | Official's given name. |
| referee3_last_name | character | Official's family name. |
| commissioner_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| commissioner_first_name | character | Official's given name. |
| commissioner_last_name | character | Official's family name. |
| venue_id | character | Arena id, stable across seasons (Uber Arena = 2 in both 2025-26 and 2026-27); a |
| venue_name | character | Name of the arena the game is played in; `venue_id` is its stable arena id. |
| venue_address | character | Postal address, German. |
| venue_capacity | integer | Arena capacity, spectators. |
| venue_is_main | logical | The club's main arena. |
| tv_channel | character | TV channel(s) showing the game, comma-joined (a list on some routes, an empty |
| tv_type | character | TV coverage type; empty or null in every sample. |
| is_free_tv | logical | Shown on free-to-air TV. |
| live_stream_url | character | Live-stream link (DBBL games link YouTube); empty when none. |
| top_score | integer | Points of the game's top scorer; null before the game. |
| top_scorer_season_player_id | character | Season-player id. BBL: a string equal to `id`. DBBL: an INTEGER that matches `/s` |
| top_scorer_real_player_id | character | Global person id: the same across clubs and seasons. Joins to `realPlayerId` every |
| top_scorer_first_name | character | Player's given name. |
| top_scorer_last_name | character | Player's family name. |

## See Also

Other SCB Functions: `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_game(game_id = "2006920"))
```

---

`scb_game_actions` **Get BBL / DBBL Game Play-by-Play**

---

## Description

Play-by-play of a finished game with shot x/y (BBL 2008-09 on, DBBL 2024-25 on).

Endpoint: `GET https://api.basketball-bundesliga.de/games/{game_id}/stats` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_game_actions(game_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| `game_id` | (required) Game id (`game_id` in `scb_games`), e.g. 2006920 (BBL) or 10130 (DBBL). One sequence per league. |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.

- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Play-by-play of a finished game with shot x/y (BBL 2008-09 on, DBBL 2024-25 on). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_game_actions`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids |
| game_id | character | Game id, a string; unique within a league only (BBL and DBBL overlap), and not cl |
| action_id | integer | Action id, an integer unique across games of a league. |
| order_id | integer | Chronological sequence of the action within the game. |
| action_identifier | character | Provider action key `<n>:<seq>` (opaque; the first part is not the period). |
| period | integer | Period: 1-4 quarters, 5+ overtimes. |
| game_time | character | Game clock as HH:MM:SS: in an action, the time LEFT in the period (00:10:00 at t |
| seconds_left | integer | Seconds left in the period at the action: `game_time` as seconds. |
| real_time | character | Wall-clock time of the action, ISO 8601 UTC. Pre-2011 games carry a placeholder: |
| type | character | Action type; see the action-type table. Records and videos use their own `type` (see t |
| qualifiers | character | Action qualifiers (shot kind, rebound side, foul kind, turnover kind); see the qualifie |
| is_successful | logical | Made / successful (shots, free throws); for other types see the action-type table. |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBBL |
| side | character | `home` or `guest` (the away side), for the team the row belongs to. |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while |
| assisting_season_player_id | character | Season-player id of the assisting player; only on assisted made shots. Joins to box-s |
| shot_x | integer | Shot x on the site's half-court chart (0-280 across, 140 through the rim); null withou |
| shot_y | integer | Shot y along the court from behind the baseline (rim near 45, arc top near 147); null |
| home_team_points | integer | Home score after this action; only on scoring actions. |
| guest_team_points | integer | Guest score after this action; only on scoring actions. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_game_actions(game_id = "2006920"))
```

`scb_game_players` **Get BBL / DBBL Game Player Box Score**

## Description

Box score of a finished game: one row per player line of both teams (2002-03 on).

Endpoint: `GET https://api.basketball-bundesliga.de/games/{game_id}/stats` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_game_players(game_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| `game_id` | (required) Game id (`game_id` in `scb_games`), e.g. 2006920 (BBL) or 10130 (DBBL). One sequence per league. |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.

- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Box score of a finished game: one row per player line of both teams (2002-03 on). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_game_players`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL ga |
| game_id | character | Game id, a string. Joins to `/games/{gameId}`. |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus |
| side | character | `home` or `guest` (the away side), for the team the row belongs to. |
| team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}` |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across season |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| shirt_number | character | Jersey number. A number or a numeric string across rows. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWARD, P |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, |
| team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Basket |
| team_tlc | character | Three-letter club code. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| seconds_played | integer | The player's playing time in SECONDS in the game. |
| points | integer | The player's points in the game. |
| two_point_shots_made | integer | The player's 2-point field goals made in the game. |
| two_point_shots_attempted | integer | The player's 2-point field goals attempted in the game. |
| two_point_shot_success_percent | double | The player's 2-point field-goal percentage in the game, 0-100 (one decimal); |
| three_point_shots_made | integer | The player's 3-point field goals made in the game. |
| three_point_shots_attempted | integer | The player's 3-point field goals attempted in the game. |
| three_point_shot_success_percent | double | The player's 3-point field-goal percentage in the game, 0-100 (one decimal); |
| field_goals_made | integer | The player's field goals made in the game. |
| field_goals_attempted | integer | The player's field goals attempted in the game. |
| field_goals_success_percent | double | The player's field-goal percentage in the game, 0-100 (one decimal); 0 when |
| free_throws_made | integer | The player's free throws made in the game. |
| free_throws_attempted | integer | The player's free throws attempted in the game. |
| free_throws_success_percent | double | The player's free-throw percentage in the game, 0-100 (one decimal); 0 when |
| offensive_rebounds | integer | The player's offensive rebounds in the game. |
| defensive_rebounds | integer | The player's defensive rebounds in the game. |
| total_rebounds | integer | The player's rebounds (offensive + defensive) in the game. |
| assists | integer | The player's assists in the game. |
| steals | integer | The player's steals in the game. |
| turnovers | integer | The player's turnovers in the game. |

| | | |
|---|---|---|
| blocks | integer | The player's blocked shots in the game. |
| fouls_committed | integer | The player's fouls committed in the game. |
| efficiency | integer | The player's efficiency (EFF = PTS + REB + AST + STL + BLK - missed FG |
| plus_minus | integer | The player's plus-minus (point margin while on court); null when the API ser |
| is_starting_five | logical | Started the game. |
| double_double | integer | The player's double-doubles in the game. |
| triple_double | integer | The player's triple-doubles in the game. |
| fouls_received | integer | The player's fouls drawn in the game. |
| opponent_team_name | character | Opponent's name. |
| field_goals_recomputed | logical | True when the API's field-goal totals disagreed with its 2-point plus 3-point p |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

## Examples

```
try(scb_game_players(game_id = "2006920"))
```

---

# scb_game_teams — Get BBL / DBBL Game Team Box Score

---

## Description

Box score of a finished game: one row per team, with the head coach and the team line.

Endpoint: `GET https://api.basketball-bundesliga.de/games/{game_id}/stats` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_game_teams(game_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| game_id | (required) Game id (game_id in scb_games), e.g. 2006920 (BBL) or 10130 (DBBL). One sequence per league. |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Box score of a finished game: one row per team, with the head coach and the team line. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_game_teams`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and |
| game_id | character | Game id, a string. Joins to `/games/{gameId}`. |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `seasc` |
| side | character | `home` or `guest` (the away side), for the team the row belongs to. |
| team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams` |
| head_coach_name | character | Name of the team's head coach. |
| status | character | Game status: PRE (scheduled), LIVE, POST (ended, not yet offici |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2 |
| season_id | integer | Season, START year (2026 = 2026-27). |
| team_name | character | Team name as listed that season (sponsor names included, e.g. 'E |
| team_tlc | character | Three-letter club code. |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPION |

| | | |
|---|---|---|
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, |
| seconds_played | integer | The team's playing time in SECONDS in the game (players plus t |
| points | integer | The team's points in the game (players plus the team line). |
| two_point_shots_made | integer | The team's 2-point field goals made in the game (players plus the |
| two_point_shots_attempted | integer | The team's 2-point field goals attempted in the game (players plus |
| two_point_shot_success_percent | double | The team's 2-point field-goal percentage in the game (players plus |
| three_point_shots_made | integer | The team's 3-point field goals made in the game (players plus the |
| three_point_shots_attempted | integer | The team's 3-point field goals attempted in the game (players plus |
| three_point_shot_success_percent | double | The team's 3-point field-goal percentage in the game (players plus |
| field_goals_made | integer | The team's field goals made in the game (players plus the team lin |
| field_goals_attempted | integer | The team's field goals attempted in the game (players plus the tea |
| field_goals_success_percent | double | The team's field-goal percentage in the game (players plus the tea |
| free_throws_made | integer | The team's free throws made in the game (players plus the team li |
| free_throws_attempted | integer | The team's free throws attempted in the game (players plus the tea |
| free_throws_success_percent | double | The team's free-throw percentage in the game (players plus the tea |
| offensive_rebounds | integer | The team's offensive rebounds in the game (players plus the team |
| defensive_rebounds | integer | The team's defensive rebounds in the game (players plus the team |
| total_rebounds | integer | The team's rebounds (offensive + defensive) in the game (players |
| assists | integer | The team's assists in the game (players plus the team line). |
| steals | integer | The team's steals in the game (players plus the team line). |
| turnovers | integer | The team's turnovers in the game (players plus the team line). |
| blocks | integer | The team's blocked shots in the game (players plus the team line). |
| fouls_committed | integer | The team's fouls committed in the game (players plus the team lin |
| efficiency | integer | The team's efficiency (EFF = PTS + REB + AST + STL + BLK - |
| plus_minus | integer | The team's point margin in the game. |
| team_line_seconds_played | integer | Team-line playing time in SECONDS (not credited to a player). |
| team_line_points | integer | Team-line points (not credited to a player). |
| team_line_two_point_shots_made | integer | Team-line 2-point field goals made (not credited to a player). |
| team_line_two_point_shots_attempted | integer | Team-line 2-point field goals attempted (not credited to a player). |
| team_line_two_point_shot_success_percent | double | Team-line 2-point field-goal percentage (not credited to a player), |
| team_line_three_point_shots_made | integer | Team-line 3-point field goals made (not credited to a player). |
| team_line_three_point_shots_attempted | integer | Team-line 3-point field goals attempted (not credited to a player). |
| team_line_three_point_shot_success_percent | double | Team-line 3-point field-goal percentage (not credited to a player), |
| team_line_field_goals_made | integer | Team-line field goals made (not credited to a player). |
| team_line_field_goals_attempted | integer | Team-line field goals attempted (not credited to a player). |
| team_line_field_goals_success_percent | double | Team-line field-goal percentage (not credited to a player), 0-100 ( |
| team_line_free_throws_made | integer | Team-line free throws made (not credited to a player). |
| team_line_free_throws_attempted | integer | Team-line free throws attempted (not credited to a player). |
| team_line_free_throws_success_percent | double | Team-line free-throw percentage (not credited to a player), 0-100 |
| team_line_offensive_rebounds | integer | Team-line offensive rebounds (not credited to a player). |
| team_line_defensive_rebounds | integer | Team-line defensive rebounds (not credited to a player). |
| team_line_total_rebounds | integer | Team-line rebounds (offensive + defensive) (not credited to a play |
| team_line_assists | integer | Team-line assists (not credited to a player). |
| team_line_steals | integer | Team-line steals (not credited to a player). |
| team_line_turnovers | integer | Team-line turnovers (not credited to a player). |
| team_line_blocks | integer | Team-line blocked shots (not credited to a player). |
| team_line_fouls_committed | integer | Team-line fouls committed (not credited to a player). |
| team_line_efficiency | integer | Team-line efficiency (EFF = PTS + REB + AST + STL + BLK - n |

| | | |
|---|---|---|
| team_line_plus_minus | integer | Team-line plus-minus: the team line has no court time, so this is n |
| field_goals_recomputed | logical | True when the API's field-goal totals disagreed with its 2-point pl |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

## Examples

```
try(scb_game_teams(game_id = "2006920"))
```

---

# scb_games — Get BBL / DBBL Games

## Description

Games of a season (the whole season in one page): teams, score by period, officials, venue and top scorer.

Endpoint: `GET https://api.basketball-bundesliga.de/games` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_games(
  league = "dbbl",
  season = NULL,
  competition = NULL,
  game_type = NULL,
  team_id = NULL,
  page_size = 500
)
```

## Arguments

| | |
|---|---|
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |
| season | (default NULL) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). Omitted = the season in progress. |

| | |
|---|---|
| competition | (default NULL) Competition filter: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CUP on the BBL host; DBBL, DBBL_CUP on the DBBL host. Omitted = all. The other host's competition answers an empty page. |
| game_type | (default NULL) Finished, scheduled or all games. Omitted = all. |
| team_id | (default NULL) Only the games of this team, home or guest (the club id equals the season-team id). |
| page_size | (default 500) Rows per page. The wrapper reads page 1 and refuses a result that needs more pages, so keep it large enough for the whole result (500 covers a season). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: wehoop_no_data (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), wehoop_bad_request (a bad argument, another 400 / 422) and wehoop_asset_fetch (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting wehoop_error. Wrappers raise; they never return an empty frame.

## Value

A wehoop_data tibble: Games of a season (the whole season in one page): teams, score by period, officials, venue and top scorer. An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_games):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBBL game id |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's season_id plus one. |

| | | |
|---|---|---|
| season_id | integer | Season, START year (2026 = 2026-27). |
| game_id | character | Game id, a string; unique within a league only (BBL and DBBL overlap), and not |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CUP on the E |
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, FIRST_ROUNI |
| status | character | Game status: PRE (scheduled), LIVE, POST (ended, not yet official), OFFICIAL |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |
| match_day | integer | Round number within the stage (Spieltag). |
| is_tbd | logical | Date or time not yet fixed. |
| is_hidden | logical | True when the API hides the game (0/1 on /games, a boolean on the header). |
| home_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| home_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| home_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| home_tlc | character | Three-letter club code. |
| home_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| home_rank | integer | The team's current table position. |
| home_streak | character | Current streak: W or L; empty when none. |
| home_streak_count | integer | Length of the current streak. |
| home_last10_victories | integer | Wins in the last 10 games. |
| home_last10_losses | integer | Losses in the last 10 games. |
| home_total_games | integer | Games the home team has played so far (the API sends it as a string). |
| guest_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| guest_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| guest_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| guest_tlc | character | Three-letter club code. |
| guest_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| guest_rank | integer | The team's current table position. |
| guest_streak | character | Current streak: W or L; empty when none. |
| guest_streak_count | integer | Length of the current streak. |
| guest_last10_victories | integer | Wins in the last 10 games. |
| guest_last10_losses | integer | Losses in the last 10 games. |
| guest_total_games | integer | Games the guest team has played so far (the API sends it as a string). |
| home_score | integer | Home team final score. |
| guest_score | integer | Guest team final score. |
| latest_period | integer | Number of the last period played (4 = Q4, 5 = OT1, ...); the API sends `Q4` on /gam |
| spectators | integer | Attendance; 0 when not recorded. |
| home_q1_score | integer | Home team points in quarter 1; 0 when quarter scores were not recorded. |
| home_q2_score | integer | Home team points in quarter 2; 0 when quarter scores were not recorded. |
| home_q3_score | integer | Home team points in quarter 3; 0 when quarter scores were not recorded. |
| home_q4_score | integer | Home team points in quarter 4; 0 when quarter scores were not recorded. |
| home_ot1_score | integer | Home team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| home_ot2_score | integer | Home team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| home_ot3_score | integer | Home team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| home_ot4_score | integer | Home team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| home_ot5_score | integer | Home team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| home_ot6_score | integer | Home team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| home_ot7_score | integer | Home team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| home_ot8_score | integer | Home team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| home_ot9_score | integer | Home team points in overtime 9; 0 when not played (null from OT5 on in old seas |

| | | |
|---|---|---|
| guest_q1_score | integer | Guest team points in quarter 1; 0 when quarter scores were not recorded. |
| guest_q2_score | integer | Guest team points in quarter 2; 0 when quarter scores were not recorded. |
| guest_q3_score | integer | Guest team points in quarter 3; 0 when quarter scores were not recorded. |
| guest_q4_score | integer | Guest team points in quarter 4; 0 when quarter scores were not recorded. |
| guest_ot1_score | integer | Guest team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| guest_ot2_score | integer | Guest team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| guest_ot3_score | integer | Guest team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| guest_ot4_score | integer | Guest team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| guest_ot5_score | integer | Guest team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| guest_ot6_score | integer | Guest team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| guest_ot7_score | integer | Guest team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| guest_ot8_score | integer | Guest team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| guest_ot9_score | integer | Guest team points in overtime 9; 0 when not played (null from OT5 on in old seas |
| referee1_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee1_first_name | character | Official's given name. |
| referee1_last_name | character | Official's family name. |
| referee2_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee2_first_name | character | Official's given name. |
| referee2_last_name | character | Official's family name. |
| referee3_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee3_first_name | character | Official's given name. |
| referee3_last_name | character | Official's family name. |
| commissioner_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| commissioner_first_name | character | Official's given name. |
| commissioner_last_name | character | Official's family name. |
| venue_id | character | Arena id, stable across seasons (Uber Arena = 2 in both 2025-26 and 2026-27); a |
| venue_name | character | Name of the arena the game is played in; `venue_id` is its stable arena id. |
| venue_address | character | Postal address, German. |
| venue_capacity | integer | Arena capacity, spectators. |
| venue_is_main | logical | The club's main arena. |
| tv_channel | character | TV channel(s) showing the game, comma-joined (a list on some routes, an empty |
| tv_type | character | TV coverage type; empty or null in every sample. |
| is_free_tv | logical | Shown on free-to-air TV. |
| live_stream_url | character | Live-stream link (DBBL games link YouTube); empty when none. |
| top_score | integer | Points of the game's top scorer; null before the game. |
| top_scorer_season_player_id | character | Season-player id. BBL: a string equal to `id`. DBBL: an INTEGER that matches `/s` |
| top_scorer_real_player_id | character | Global person id: the same across clubs and seasons. Joins to `realPlayerId` every |
| top_scorer_first_name | character | Player's given name. |
| top_scorer_last_name | character | Player's family name. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_games())
```

---

`scb_games_recent` **Get BBL / DBBL Recent Games**

---

## Description

Recent, live and upcoming games, one row per game with its past / present / future bucket.

Endpoint: `GET https://api.basketball-bundesliga.de/games/recent` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_games_recent(league = "dbbl", slot_size = NULL)
```

## Arguments

| | |
|---|---|
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |
| `slot_size` | (default `NULL`) Games per bucket (past, present, future); the API's default is 9. |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.

- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Recent, live and upcoming games, one row per game with its past / present / future bucket. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_games_recent`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ic |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| game_id | character | Game id, a string; unique within a league only (BBL and DBBL overlap), and not |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CUP on the E |
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, FIRST_ROUNI |
| status | character | Game status: PRE (scheduled), LIVE, POST (ended, not yet official), OFFICIAL |
| bucket | character | Which list of `/games/recent` the game came from: `past` (finished), `present` (tod |
| game_progress | character | Live progress code: E (ended), empty or null before the game. Meaning inferred. |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |
| match_day | integer | Round number within the stage (Spieltag). |
| is_tbd | logical | Date or time not yet fixed. |
| is_hidden | logical | True when the API hides the game (0/1 on /games, a boolean on the header). |
| home_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| home_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| home_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| home_tlc | character | Three-letter club code. |
| home_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| home_rank | integer | The team's current table position. |
| home_streak | character | Current streak: W or L; empty when none. |
| home_streak_count | integer | Length of the current streak. |
| home_last10_victories | integer | Wins in the last 10 games. |
| home_last10_losses | integer | Losses in the last 10 games. |
| home_total_games | integer | Games the home team has played so far (the API sends it as a string). |
| guest_team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| guest_season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBB |
| guest_team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE Baskets Old |
| guest_tlc | character | Three-letter club code. |
| guest_is_foreign | logical | The team is not in this league (cup opponents from the ProA / ProB). |
| guest_rank | integer | The team's current table position. |
| guest_streak | character | Current streak: W or L; empty when none. |
| guest_streak_count | integer | Length of the current streak. |
| guest_last10_victories | integer | Wins in the last 10 games. |
| guest_last10_losses | integer | Losses in the last 10 games. |
| guest_total_games | integer | Games the guest team has played so far (the API sends it as a string). |

| | | |
|---|---|---|
| home_score | integer | Home team final score. |
| guest_score | integer | Guest team final score. |
| latest_period | integer | Number of the last period played (4 = Q4, 5 = OT1, ...); the API sends `Q4` on /gam |
| spectators | integer | Attendance; 0 when not recorded. |
| home_q1_score | integer | Home team points in quarter 1; 0 when quarter scores were not recorded. |
| home_q2_score | integer | Home team points in quarter 2; 0 when quarter scores were not recorded. |
| home_q3_score | integer | Home team points in quarter 3; 0 when quarter scores were not recorded. |
| home_q4_score | integer | Home team points in quarter 4; 0 when quarter scores were not recorded. |
| home_ot1_score | integer | Home team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| home_ot2_score | integer | Home team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| home_ot3_score | integer | Home team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| home_ot4_score | integer | Home team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| home_ot5_score | integer | Home team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| home_ot6_score | integer | Home team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| home_ot7_score | integer | Home team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| home_ot8_score | integer | Home team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| home_ot9_score | integer | Home team points in overtime 9; 0 when not played (null from OT5 on in old seas |
| guest_q1_score | integer | Guest team points in quarter 1; 0 when quarter scores were not recorded. |
| guest_q2_score | integer | Guest team points in quarter 2; 0 when quarter scores were not recorded. |
| guest_q3_score | integer | Guest team points in quarter 3; 0 when quarter scores were not recorded. |
| guest_q4_score | integer | Guest team points in quarter 4; 0 when quarter scores were not recorded. |
| guest_ot1_score | integer | Guest team points in overtime 1; 0 when not played (null from OT5 on in old seas |
| guest_ot2_score | integer | Guest team points in overtime 2; 0 when not played (null from OT5 on in old seas |
| guest_ot3_score | integer | Guest team points in overtime 3; 0 when not played (null from OT5 on in old seas |
| guest_ot4_score | integer | Guest team points in overtime 4; 0 when not played (null from OT5 on in old seas |
| guest_ot5_score | integer | Guest team points in overtime 5; 0 when not played (null from OT5 on in old seas |
| guest_ot6_score | integer | Guest team points in overtime 6; 0 when not played (null from OT5 on in old seas |
| guest_ot7_score | integer | Guest team points in overtime 7; 0 when not played (null from OT5 on in old seas |
| guest_ot8_score | integer | Guest team points in overtime 8; 0 when not played (null from OT5 on in old seas |
| guest_ot9_score | integer | Guest team points in overtime 9; 0 when not played (null from OT5 on in old seas |
| referee1_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee1_first_name | character | Official's given name. |
| referee1_last_name | character | Official's family name. |
| referee2_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee2_first_name | character | Official's given name. |
| referee2_last_name | character | Official's family name. |
| referee3_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| referee3_first_name | character | Official's given name. |
| referee3_last_name | character | Official's family name. |
| commissioner_id | character | Official id; joins to /referees `refId` (= its `id`). /referees lists only the current offici |
| commissioner_first_name | character | Official's given name. |
| commissioner_last_name | character | Official's family name. |
| venue_id | character | Arena id, stable across seasons (Uber Arena = 2 in both 2025-26 and 2026-27); a c |
| venue_name | character | Name of the arena the game is played in; `venue_id` is its stable arena id. |
| venue_address | character | Postal address, German. |
| venue_capacity | integer | Arena capacity, spectators. |
| venue_is_main | logical | The club's main arena. |
| tv_channel | character | TV channel(s) showing the game, comma-joined (a list on some routes, an empty |

| | | |
|---|---|---|
| tv_type | character | TV coverage type; empty or null in every sample. |
| is_free_tv | logical | Shown on free-to-air TV. |
| live_stream_url | character | Live-stream link (DBBL games link YouTube); empty when none. |
| top_score | integer | Points of the game's top scorer; null before the game. |
| top_scorer_season_player_id | character | Season-player id. BBL: a string equal to `id`. DBBL: an INTEGER that matches /s |
| top_scorer_real_player_id | character | Global person id: the same across clubs and seasons. Joins to `realPlayerId` every |
| top_scorer_first_name | character | Player's given name. |
| top_scorer_last_name | character | Player's family name. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_games_recent())
```

---

# scb_player_statistics Get BBL / DBBL Season Player Statistics

---

## Description

Season player stat table: totals and per-game values, one row per player and stat block.

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}/player-statistics` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_player_statistics(season, league = "dbbl", display_type = "MAIN_ROUND")
```

## Arguments

| | |
|---|---|
| `season` | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |
| `display_type` | (default `"MAIN_ROUND"`) Which games the table covers: MAIN_ROUND (regular season), PLAYOFFS, PLAYOFFS_TOTAL, MAIN_ROUND_AND_PLAY_OFFS or CUP. |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Season player stat table: totals and per-game values, one row per player and stat block. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_player_statistics`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBl |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` |
| display_type | character | Which games the table covers, echoed from the request (MAIN_ROUN |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across s |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlay` |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| shirt_number | character | Jersey number of the player, kept as text so a `00` jersey survives. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWAI |
| team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{te` |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2 |
| team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE B |
| team_tlc | character | Three-letter club code. |

| | | |
|---|---|---|
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); season |
| round_code | integer | Provider round code; 1 for MAIN_ROUND in every sample (mapping |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CU |
| seconds_played | integer | The player's season total playing time in SECONDS over displayType. |
| points | integer | The player's season total points over displayType. |
| two_point_shots_made | integer | The player's season total 2-point field goals made over displayType. |
| two_point_shots_attempted | integer | The player's season total 2-point field goals attempted over displayType |
| two_point_shot_success_percent | double | The player's season total 2-point field-goal percentage over displaytype |
| three_point_shots_made | integer | The player's season total 3-point field goals made over displayType. |
| three_point_shots_attempted | integer | The player's season total 3-point field goals attempted over displayType |
| three_point_shot_success_percent | double | The player's season total 3-point field-goal percentage over displaytype |
| field_goals_made | integer | The player's season total field goals made over displayType. |
| field_goals_attempted | integer | The player's season total field goals attempted over displayType. |
| field_goals_success_percent | double | The player's season total field-goal percentage over displaytype, 0-100 |
| free_throws_made | integer | The player's season total free throws made over displayType. |
| free_throws_attempted | integer | The player's season total free throws attempted over displayType. |
| free_throws_success_percent | double | The player's season total free-throw percentage over displaytype, 0-100 |
| offensive_rebounds | integer | The player's season total offensive rebounds over displayType. |
| defensive_rebounds | integer | The player's season total defensive rebounds over displayType. |
| total_rebounds | integer | The player's season total rebounds (offensive + defensive) over display |
| assists | integer | The player's season total assists over displayType. |
| steals | integer | The player's season total steals over displayType. |
| turnovers | integer | The player's season total turnovers over displayType. |
| blocks | integer | The player's season total blocked shots over displayType. |
| fouls_committed | integer | The player's season total fouls committed over displayType. |
| efficiency | integer | The player's season total efficiency (EFF = PTS + REB + AST + STL - |
| plus_minus | integer | The player's season total plus-minus over displaytype. |
| games_played | integer | The player's season total games played over displayType. |
| seconds_per_game | double | Per-game average of seconds, seconds. |
| points_per_game | double | Per-game average of points. |
| two_point_shots_made_per_game | double | Per-game average of 2-point field goals made. |
| two_point_shots_attempted_per_game | double | Per-game average of 2-point field goals attempted. |
| three_point_shots_made_per_game | double | Per-game average of 3-point field goals made. |
| three_point_shots_attempted_per_game | double | Per-game average of 3-point field goals attempted. |
| field_goals_made_per_game | double | Per-game average of field goals made. |
| field_goals_attempted_per_game | double | Per-game average of field goals attempted. |
| field_goals_success_percent_per_game | double | Per-game average of field-goal percentage (0 in every sample: unusable |
| free_throws_made_per_game | double | Per-game average of free throws made. |
| free_throws_attempted_per_game | double | Per-game average of free throws attempted. |
| defensive_rebounds_per_game | double | Per-game average of defensive rebounds. |
| offensive_rebounds_per_game | double | Per-game average of offensive rebounds. |
| total_rebounds_per_game | double | Per-game average of rebounds (offensive + defensive). |
| assists_per_game | double | Per-game average of assists. |
| steals_per_game | double | Per-game average of steals. |
| turnovers_per_game | double | Per-game average of turnovers. |
| blocks_per_game | double | Per-game average of blocked shots. |
| fouls_committed_per_game | double | Per-game average of fouls committed. |
| efficiency_per_game | double | Per-game average of efficiency (EFF = PTS + REB + AST + STL + BL |

| | | |
|---|---|---|
| plus_minus_per_game | double | Per-game average of plus-minus (point margin while on court). |
| double_double | integer | The player's season total double-doubles over displayType. |
| double_double_per_game | double | Per-game average of double-doubles. |
| games_started | integer | The player's season total games started over displayType. |
| triple_double | integer | The player's season total triple-doubles over displayType. |
| triple_double_per_game | double | Per-game average of triple-doubles. |
| fouls_received | integer | The player's season total fouls drawn over displayType. |
| fouls_received_per_game | double | Per-game average of fouls drawn. |
| overall | logical | False in every sample (meaning unverified). |
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, FIRS |
| aggregation | character | Aggregation label as sent: STAGE in player tables, MAIN_ROUND in |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

## Examples

```
try(scb_player_statistics(season = 2026))
```

---

# scb_playoff_games Get BBL / DBBL Playoff Games

---

## Description

Every playoff game of a season with its series and the points of each series team.

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}/playoff-tree` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_playoff_games(season, league = "dbbl")
```

## Arguments

| | |
|---|---|
| season | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_playoff_games(season = 2026)
```

## Value

A `wehoop_data` tibble: Every playoff game of a season with its series and the points of each series team. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_playoff_games`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, so |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); `season` minus one. |
| round | character | Round of the bracket: `quarter_finals`, `semi_finals` or `finals`. |
| series | integer | 1-based position of the game's series within its round, in the order the API lists them. |
| game_id | character | Game id, a string (the playoff tree sends an integer); joins `scb_games.game_id`. |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |
| timestamp | integer | Tip-off, Unix seconds. |
| game_winner_id | character | Season-team id of the game winner, a string (the playoff tree sends an integer). |
| home_team_id | character | Season-team id of the home team, a string (the playoff tree sends an integer). |
| guest_team_id | character | Season-team id of the guest (away) team, a string (the playoff tree sends an integer). |

| | | |
|---|---|---|
| team_a_id | character | Season-team id of the series' team A (the higher seed); `points_team_a` is its score, whichever |
| team_b_id | character | Season-team id of the series' team B; `points_team_b` is its score, whichever side hosted. |
| points_team_a | integer | Team A's points in the game. |
| points_team_b | integer | Team B's points in the game. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

---

`scb_playoff_series` **Get BBL / DBBL Playoff Series**

---

## Description

Playoff bracket of a season: one row per series (quarter-finals, semi-finals, finals).

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}/playoff-tree` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_playoff_series(season, league = "dbbl")
```

## Arguments

| | |
|---|---|
| `season` | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.

- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_playoff_series(season = 2026)
```

## Value

A `wehoop_data` tibble: Playoff bracket of a season: one row per series (quarter-finals, semi-finals, finals). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_playoff_series`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| round | character | Round of the bracket: `quarter_finals`, `semi_finals` or `finals`. |
| series | integer | 1-based position of the series within its round, in the order the API lists them. |
| match_winner_id | character | Season-team id of the series winner, a string (the playoff tree sends an integer). |
| is_match_finished | logical | The series is decided. |
| team_a_id | character | Season-team id of series team A (the higher seed is A), a string (the playoff tree sends it as a |
| team_a_name | character | Name of series team A (the higher seed is A) as listed that season. |
| team_a_tlc | character | Three-letter club code of series team A (the higher seed is A). |
| team_a_rank | integer | Regular-season table position (the seed) of series team A (the higher seed is A). |
| team_a_victories | integer | Series wins of team A. |
| team_b_id | character | Season-team id of series team B (the higher seed is A), a string (the playoff tree sends it as a |
| team_b_name | character | Name of series team B (the higher seed is A) as listed that season. |
| team_b_tlc | character | Three-letter club code of series team B (the higher seed is A). |
| team_b_rank | integer | Regular-season table position (the seed) of series team B (the higher seed is A). |
| team_b_victories | integer | Series wins of team B. |
| games | integer | Games of the series. |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

---

# scb_records_per_game Get BBL / DBBL Single-Game Records

---

## Description

Single-game records: the season's and the all-time ones, one row per record.

Endpoint: GET https://api.basketball-bundesliga.de/records/per-game/{season} (DBBL: the same route on https://api-1.dbbl.scb.world).

## Usage

```
scb_records_per_game(season, league = "dbbl")
```

## Arguments

| | |
|---|---|
| season | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.

- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_records_per_game(season = 2026)
```

## Value

A `wehoop_data` tibble: Single-game records: the season's and the all-time ones, one row per record. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_records_per_game`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, s |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); `season` minus one. |
| scope | character | `season` for the requested season's records, `all_time` for the all-time list. |
| stat | character | The record's stat: points, rebounds, assists, steals, blocks, turnovers, efficiency, free_throws_r |
| place | integer | Rank within the list (1 = best); 0 on some rows. |
| all_time_place | integer | Rank on the all-time list; null when not on it. |
| order_id | integer | Sort order within the list. |
| all_time_order_id | integer | Sort order on the all-time list; null when not on it. |
| value | double | The record value. A STRING in /records/per-game ('52'); a number in /records/season-averag |
| ratio | double | [made, attempted] as strings, on percentage records only; null otherwise. |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he stays, |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` everywhere. |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| team_name | character | Team name as listed that season (sponsor names included). |
| team_tlc | character | The player's team code. |
| home_tlc | character | Three-letter code of the home team of the record game. |
| guest_tlc | character | Three-letter code of the guest team of the record game. |
| game_id | character | Game id, a string. Joins to `/games/{gameId}`. |
| scheduled_time | character | Scheduled tip-off, ISO 8601 UTC. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`,

scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

---

scb_records_season_averages

**Get BBL / DBBL Season-Average Records**

---

## Description

All-time season-average records (points, rebounds, assists, ...), one row per record.

Endpoint: `GET https://api.basketball-bundesliga.de/records/season-averages` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_records_season_averages(league = "dbbl")
```

## Arguments

| | |
|---|---|
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.

- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_records_season_averages()
```

## Value

A `wehoop_data` tibble: All-time season-average records (points, rebounds, assists, ...), one row per record. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_records_season_averages`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, s |
| stat | character | The record's stat list: assist_average, block_average, efficiency_average, foul_average, free_th |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| place | integer | Rank within the list (1 = best); 0 on some rows. |
| order_id | integer | Sort order within the list. |
| value | double | The record value. A STRING in /records/per-game ('52'); a number in /records/season-averag |
| ratio | double | [made, attempted] as strings, on percentage records only; null otherwise. |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he stays, |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` everywhere. |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| team_name | character | Team name as listed that season (sponsor names included). |
| team_tlc | character | The player's team code. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

---

# scb_referees — Get BBL Referees

## Description

Current referees and commissioners (BBL only).

Endpoint: `GET https://api.basketball-bundesliga.de/referees`

## Usage

```
scb_referees(page_size = 100)
```

## Arguments

| | |
|---|---|
| page_size | (default 100) Rows per page. 100 returns all current officials in one page; a result that needs more pages is refused. |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: wehoop_no_data (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), wehoop_bad_request (a bad argument, another 400 / 422) and wehoop_asset_fetch (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting wehoop_error. Wrappers raise; they never return an empty frame.

```
scb_referees()
```

## Value

A wehoop_data tibble: Current referees and commissioners (BBL only). An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_referees):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBBL game ids overlap, sc |
| referee_id | character | Official id (refId, equal to the API's id); joins referee1_id .. commissioner_id of scb_gam |
| first_name | character | Official's given name. |

| | | |
|---|---|---|
| last_name | character | Official's family name. |
| official_function | character | REFEREE or COMMISSIONER. |
| participations | integer | League games officiated. |
| year_joined_bbl | integer | Year the official joined the league. |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

---

# scb_season_player Get BBL / DBBL Season Player Profile

---

## Description

Player profile by season-player id: bio, position, club and previous clubs.

Endpoint: GET https://api.basketball-bundesliga.de/season-players/{season_player_id} (DBBL: the same route on https://api-1.dbbl.scb.world).

## Usage

```
scb_season_player(season_player_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| season_player_id | (required) Season-player id (season_player_id in a box score, an action or scb_season_players): one player's engagement with one club. |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.

- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_season_player(season_player_id = "56414")
```

## Value

A `wehoop_data` tibble: Player profile by season-player id: bio, position, club and previous clubs. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_season_player`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids ov |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` everywhe |
| active_season_player_id | character | The player's current season-player id (his latest club engagement). Joins to `/season-p` |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| birth_date | character | Date of birth, YYYY-MM-DD. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWARD, POWER_F |
| shirt_number | character | Jersey number. A number or a numeric string across rows. |
| height | double | Height in metres (float32 noise: 1.77 arrives as 1.7699999809265137). |
| weight | integer | Weight in kg; 0 when unknown. |
| nationalities | character | Nationalities, ISO 3166-1 alpha-2 codes. |
| is_all_star | logical | All-Star selection flag. |
| is_young | logical | Young-player flag (German-trained youth rule; inferred). |
| has_changed_teams | logical | Moved clubs during the season. |
| bbl_games_played | integer | Career league games played (BBL host; 0 on DBBL). |
| team_id | character | Club id, a string; equal to `season_team_id` in every sample. Joins to `scb_team` and `sc` |
| team_name | character | Team name as listed that season (sponsor names included). |
| previous_teams | character | Former clubs, free text (German names). |
| season_ids | character | Seasons (start years) with this player or club. |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

---

`scb_season_players` **Get BBL / DBBL Season Players**

---

## Description

Every player registered in a season: the id crosswalk (season_player_id, real_player_id, season_team_id).

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_season_players(season, league = "dbbl")
```

## Arguments

| | |
|---|---|
| `season` | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.

- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Every player registered in a season: the id crosswalk (season_player_id, real_player_id, season_team_id). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_season_players`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, s |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); `season` minus one. |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he stays, |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` everywhere. |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBBL 2024-202 |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_season_players(season = 2026))
```

---

`scb_season_teams` **Get BBL / DBBL Season Teams**

---

## Description

Every team of a season: club id, season-team id, name and three-letter code.

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_season_teams(season, league = "dbbl")
```

## Arguments

| | |
|---|---|
| `season` | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A wehoop_data tibble: Every team of a season: club id, season-team id, name and three-letter code. An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_season_teams):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBBL game ids overlap, so |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's season_id plus one. |
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); season minus one. |
| team_id | character | Club id; equal to seasonTeamId in every sample. Joins to /teams/{teamId}. |
| season_team_id | character | Season-team id; equal to the club teamId in every sample (BBL 2005-2026, DBBL 2024-2026 |
| team_name | character | Team name as listed that season (sponsor names included). |
| tlc | character | Three-letter club code. |
| logo_url | character | Logo image URL (per season). |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

## Examples

```
try(scb_season_teams(season = 2026))
```

---

# scb_seasons — Get BBL / DBBL Seasons

## Description

Seasons, newest first, with the active flag (BBL back to 1966, DBBL 2024 on).

Endpoint: GET https://api.basketball-bundesliga.de/seasons (DBBL: the same route on https://api-1.dbbl.scb.world).

## Usage

```
scb_seasons(league = "dbbl")
```

## Arguments

| | |
|---|---|
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: Seasons, newest first, with the active flag (BBL back to 1966, DBBL 2024 on). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_seasons`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, so a gam |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | The API's own season key, the START year (2025 = 2025-26); `season` minus one. |
| is_active | logical | True for the season in progress. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_seasons())
```

---

`scb_standings` **Get BBL / DBBL Standings**

---

## Description

League table of a season: record, home / away split, streaks and tie-break detail (2005-06 on).

Endpoint: `GET https://api.basketball-bundesliga.de/standings` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_standings(league = "dbbl", season = NULL, competition = NULL)
```

## Arguments

| | |
|---|---|
| `league` | (default `"dbbl"`, the women's DBBL) League: `bbl` (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or `dbbl` (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |
| `season` | (default `NULL`) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). Omitted = the season in progress. Seasons before about 2000 answer an empty table. |
| `competition` | (default `NULL`) Competition: BBL, BBL_CUP, ALLSTAR, CHAMPIONS_CUP (BBL host) or DBBL, DBBL_CUP (DBBL host). BBL_CUP answers an empty table. |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.

- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

## Value

A `wehoop_data` tibble: League table of a season: record, home / away split, streaks and tie-break detail (2005-06 on). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_standings`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL gan |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus o |
| standing_id | character | Opaque id of the standings row, a string. |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, D |
| team_id | character | Club id; equal to `seasonTeamId` in every sample. Joins to `/teams/{teamId}`. |
| team_name | character | Team name; null on some rows. |
| team_tlc | character | Three-letter club code. |
| team_name_short | character | Short team name (null on DBBL). |
| has_reached_playoffs | logical | Clinched a playoff place. |
| has_reached_playins | logical | Clinched at least the play-ins. |
| is_relegation_certain | logical | Relegated for certain. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CUP on t |
| stage | character | Stage: MAIN_ROUND (Hauptrunde, regular season), PLAYINS, FIRST_RO |
| rank | integer | Table position. |
| streak | character | Current streak: W or L; empty when none. |
| streak_count | integer | Length of the current streak. |
| point_corrections | character | Table-point deductions; [] in every sample. |
| home_victories | integer | Home games: wins. |
| home_losses | integer | Home games: number of losses in the table. |
| home_victory_percentage | double | Home games: win percentage, 0-100. |
| away_victories | integer | Away games: wins. |
| away_losses | integer | Away games: number of losses in the table. |
| away_victory_percentage | double | Away games: win percentage, 0-100. |
| table_points | integer | Table points scored (2 per win), from the API's `totalPointsMade`; the provid |
| table_points_against | integer | Table points conceded (2 per loss), from the API's `totalPointsAgainst`; per |
| total_victory_percentage | double | All games: win percentage, 0-100. |
| total_games | integer | Games played in the table (home and away together). |
| total_victories | integer | All games: wins. |

| | | |
|---|---|---|
| total_losses | integer | Losses in the table over all games. |
| last10_victories | integer | Wins in the last 10 games. |
| last10_losses | integer | Losses in the last 10 games. |
| last10_percentage | double | Win percentage over the last 10 games, 0-100. |
| last10_outcomes | character | Results of the last games, newest first: W or L. |
| home_last10_victories | integer | Home games: last 10 games, wins. |
| home_last10_losses | integer | Home games: last 10 games, losses. |
| home_last10_victory_percentage | double | Home games: last 10 games, win percentage, 0-100. |
| home_streak | character | Home games: current streak, W or L. |
| home_streak_count | integer | Home games: length of the current streak. |
| away_last10_victories | integer | Away games: last 10 games, wins. |
| away_last10_losses | integer | Away games: last 10 games, losses. |
| away_last10_victory_percentage | double | Away games: last 10 games, win percentage, 0-100. |
| away_streak | character | Away games: current streak, W or L. |
| away_streak_count | integer | Away games: length of the current streak. |
| updated_date | character | Last update of the row, ISO 8601 UTC. |
| ranking_comparison_info | character | Tie-break detail among teams level on record. |
| home_points_for | integer | Home games: basketball points scored (the API's football-style `GoalsMade`). |
| away_points_for | integer | Away games: basketball points scored (the API's football-style `GoalsMade`). |
| total_points_for | integer | All games: basketball points scored (the API's football-style `GoalsMade`). |
| home_points_against | integer | Home games: basketball points allowed (the API's football-style `GoalsAgain` |
| away_points_against | integer | Away games: basketball points allowed (the API's football-style `GoalsAgains` |
| total_points_against | integer | All games: basketball points allowed (the API's football-style `GoalsAgainst`) |
| home_point_difference | integer | Home games: point differential, scored minus allowed (the API's `GoalDiffer` |
| away_point_difference | integer | Away games: point differential, scored minus allowed (the API's `GoalDiffer` |
| total_point_difference | integer | All games: point differential, scored minus allowed (the API's `GoalDifferen` |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_standings())
```

---

`scb_team` **Get BBL / DBBL Team Profile**

---

## Description

Club profile: names, code, home page, ticket shop, year joined and honours.

Endpoint: `GET https://api.basketball-bundesliga.de/teams/{team_id}` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_team(team_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| team_id | (required) Club id (team_id in scb_season_teams), e.g. 413 (ALBA BERLIN). |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: wehoop_no_data (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), wehoop_bad_request (a bad argument, another 400 / 422) and wehoop_asset_fetch (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting wehoop_error. Wrappers raise; they never return an empty frame.

```
scb_team(team_id = "413")
```

## Value

A wehoop_data tibble: Club profile: names, code, home page, ticket shop, year joined and honours. An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_team):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, |
| team_id | character | Club id, a string; equal to `season_team_id` in every sample. Joins to `scb_team` and `scb_sea` |
| tlc | character | Three-letter club code. |
| name_full | character | Full club name as the API lists it today. |
| name_short | character | Short name (often the city); null on DBBL. |
| mascot | character | Club mascot name (null when none). |
| homepage_url | character | Club website address as sent (it can lack the scheme). |
| ticket_url | character | Ticket shop URL. |
| year_joined_bbl | integer | Year the club joined the league. |
| order_id | integer | Display order of the club on the site; null on some clubs. |
| best_achievements | character | Honours, German free text (Deutscher Meister = German champion, Pokalsieger = cup winn |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

---

# `scb_team_rosters` Get BBL Team Rosters

---

## Description

Per club, the current roster, one row per player (BBL only).

Endpoint: `GET https://api.basketball-bundesliga.de/transfers/teams`

## Usage

```
scb_team_rosters()
```

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.

- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_team_rosters()
```

## Value

A `wehoop_data` tibble: Per club, the current roster, one row per player (BBL only). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_team_rosters`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids overlap, s |
| team_id | character | Club id, a string; equal to `season_team_id` in every sample. Joins to `scb_team` and `scb_seas` |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| season_team_id | character | Season-team id; equal to the club `teamId` in every sample (BBL 2005-2026, DBBL 2024-202 |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he stays, |
| real_player_id | integer | Global person id: the same across clubs and seasons. Joins to `realPlayerId` everywhere. |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| birth_date | character | Date of birth, YYYY-MM-DD. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWARD, POWER_FORWA |
| shirt_number | character | Jersey number as sent (a string: it can be `00`). |
| height | double | Height in metres (float32 noise: 1.77 arrives as 1.7699999809265137). |
| weight | integer | Weight in kg; 0 when unknown. |
| nationalities | character | Nationalities, ISO 3166-1 alpha-2 codes. |
| is_all_star | logical | All-Star selection flag. |
| is_young | logical | Young-player flag (German-trained youth rule; inferred). |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`,

scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_seasons(), scb_team_statistics(), scb_transfers(), scb_transfers_teams()

---

scb_team_seasons **Get BBL / DBBL Team Seasons**

---

## Description

Every season a club played: its name that season and its championship and cup titles.

Endpoint: `GET https://api.basketball-bundesliga.de/teams/{team_id}` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_team_seasons(team_id, league = "dbbl")
```

## Arguments

| | |
|---|---|
| team_id | (required) Club id (team_id in scb_season_teams), e.g. 413 (ALBA BERLIN). |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.

- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_team_seasons(team_id = "413")
```

## Value

A `wehoop_data` tibble: Every season a club played: its name that season and its championship and cup titles. An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_team_seasons`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids ov |
| team_id | character | Club id, a string; equal to `season_team_id` in every sample. Joins to `scb_team` and s |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's `season_id` plus one. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| season_team_id | character | Season-team id, a string; equal to the club `team_id` in every sample (BBL 2005-2026, |
| team_name | character | Team name as listed that season (sponsor names included). |
| is_championship_winner | logical | Won the championship that season. |
| is_cup_winner | logical | Won the cup that season. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_statistics()`, `scb_transfers()`, `scb_transfers_teams()`

---

# `scb_team_statistics` Get BBL / DBBL Season Team Statistics

---

## Description

Season team stat table: totals and per-game values, one row per team and stat block.

Endpoint: `GET https://api.basketball-bundesliga.de/seasons/{season}/team-statistics` (DBBL: the same route on `https://api-1.dbbl.scb.world`).

## Usage

```
scb_team_statistics(season, league = "dbbl", display_type = "MAIN_ROUND")
```

## Arguments

| | |
|---|---|
| season | (required) Season END year (2026 = 2025-26, the sdv key). Sent to the API as its START-year seasonId (season - 1). |
| league | (default "dbbl", the women's DBBL) League: bbl (men, easyCredit Basketball Bundesliga, api.basketball-bundesliga.de) or dbbl (women, Toyota Damen-Basketball-Bundesliga, api-1.dbbl.scb.world). The two share one API; game ids overlap, so a game is (league, game_id). |
| display_type | (default "MAIN_ROUND") Which games the table covers: MAIN_ROUND (regular season), PLAYOFFS, PLAYOFFS_TOTAL, MAIN_ROUND_AND_PLAY_OFFS or CUP. |

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (Inhalte) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant x-api-key and an x-api-secret that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, : turned into -) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with wehoop_asset_fetch.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year season is sent as season - 1. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: wehoop_no_data (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), wehoop_bad_request (a bad argument, another 400 / 422) and wehoop_asset_fetch (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting wehoop_error. Wrappers raise; they never return an empty frame.

## Value

A wehoop_data tibble: Season team stat table: totals and per-game values, one row per team and stat block. An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_team_statistics):

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBl |
| season | integer | Season END year (2026 = 2025-26), the sdv key; the API's season_id |
| display_type | character | Which games the table covers, echoed from the request (MAIN_ROUN |
| team_id | character | Club id; equal to seasonTeamId in every sample. Joins to /teams/{te |
| season_team_id | character | Season-team id; equal to the club teamId in every sample (BBL 2005-2 |
| team_name | character | Team name as listed that season (sponsor names included, e.g. 'EWE B |
| team_tlc | character | Three-letter club code. |
| season_id | integer | Season, START year (2026 = 2026-27). |
| round_code | integer | Provider round code; 1 for MAIN_ROUND in every sample (mapping |
| competition | character | Competition: BBL, BBL_CUP (Pokal), ALLSTAR, CHAMPIONS_CU |
| aggregation | character | Aggregation label as sent: STAGE in player tables, MAIN_ROUND in |
| seconds_played | integer | The team's season total playing time in SECONDS over displayType. |
| points | integer | The team's season total points over displayType. |
| two_point_shots_made | integer | The team's season total 2-point field goals made over displayType. |
| two_point_shots_attempted | integer | The team's season total 2-point field goals attempted over displayType. |
| two_point_shot_success_percent | double | The team's season total 2-point field-goal percentage over displaytype, |
| three_point_shots_made | integer | The team's season total 3-point field goals made over displayType. |
| three_point_shots_attempted | integer | The team's season total 3-point field goals attempted over displayType. |
| three_point_shot_success_percent | double | The team's season total 3-point field-goal percentage over displaytype, |
| field_goals_made | integer | The team's season total field goals made over displayType. |
| field_goals_attempted | integer | The team's season total field goals attempted over displayType. |
| field_goals_success_percent | double | The team's season total field-goal percentage over displaytype, 0-100 ( |
| free_throws_made | integer | The team's season total free throws made over displayType. |
| free_throws_attempted | integer | The team's season total free throws attempted over displayType. |
| free_throws_success_percent | double | The team's season total free-throw percentage over displaytype, 0-100 ( |
| offensive_rebounds | integer | The team's season total offensive rebounds over displayType. |
| defensive_rebounds | integer | The team's season total defensive rebounds over displayType. |
| total_rebounds | integer | The team's season total rebounds (offensive + defensive) over displayT |
| assists | integer | The team's season total assists over displayType. |
| steals | integer | The team's season total steals over displayType. |
| turnovers | integer | The team's season total turnovers over displayType. |
| blocks | integer | The team's season total blocked shots over displayType. |
| fouls_committed | integer | The team's season total fouls committed over displayType. |
| efficiency | integer | The team's season total efficiency (EFF = PTS + REB + AST + STL + |
| plus_minus | integer | The team's season total plus-minus over displaytype. |
| games_played | integer | The team's season total games played over displayType. |
| seconds_per_game | double | Per-game average of seconds, seconds. |
| points_per_game | double | Per-game average of points. |
| two_point_shots_made_per_game | double | Per-game average of 2-point field goals made. |
| two_point_shots_attempted_per_game | double | Per-game average of 2-point field goals attempted. |
| three_point_shots_made_per_game | double | Per-game average of 3-point field goals made. |
| three_point_shots_attempted_per_game | double | Per-game average of 3-point field goals attempted. |
| field_goals_made_per_game | double | Per-game average of field goals made. |
| field_goals_attempted_per_game | double | Per-game average of field goals attempted. |
| field_goals_success_percent_per_game | double | Per-game average of field-goal percentage (0 in every sample: unusable |
| free_throws_made_per_game | double | Per-game average of free throws made. |
| free_throws_attempted_per_game | double | Per-game average of free throws attempted. |
| defensive_rebounds_per_game | double | Per-game average of defensive rebounds. |

| | | |
|---|---|---|
| offensive_rebounds_per_game | double | Per-game average of offensive rebounds. |
| total_rebounds_per_game | double | Per-game average of rebounds (offensive + defensive). |
| assists_per_game | double | Per-game average of assists. |
| steals_per_game | double | Per-game average of steals. |
| turnovers_per_game | double | Per-game average of turnovers. |
| blocks_per_game | double | Per-game average of blocked shots. |
| fouls_committed_per_game | double | Per-game average of fouls committed. |
| efficiency_per_game | double | Per-game average of efficiency (EFF = PTS + REB + AST + STL + BL |
| plus_minus_per_game | double | Per-game average of plus-minus (point margin while on court). |
| double_double | integer | The team's season total double-doubles over displayType. |
| double_double_per_game | double | Per-game average of double-doubles. |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_transfers()`, `scb_transfers_teams()`

## Examples

```
try(scb_team_statistics(season = 2026))
```

---

`scb_transfers` **Get BBL Transfers**

---

## Description

Latest transfers, one row per move (BBL only).

Endpoint: `GET https://api.basketball-bundesliga.de/transfers`

## Usage

```
scb_transfers()
```

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.

- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_transfers()
```

## Value

A `wehoop_data` tibble: Latest transfers, one row per move (BBL only). An empty valid answer raises `wehoop_no_data`; the columns are always these, in this order (sdv-py `scb_transfers`):

| col_name | type | description |
|---|---|---|
| league | character | League the row belongs to: `bbl` (men) or `dbbl` (women); BBL and DBBL game ids ove |
| transfer_date | character | Date of the transfer, YYYY-MM-DD (the API's `sortingDate`). |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he |
| real_player_id | integer | Global person id (integer): the same player across clubs and seasons. Join players acros |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWARD, POWER_FC |
| source_team_id | character | Club id the player left; null when the source is not a league club. |
| source_team_name | character | Name of the club the player left (the free-text field, else the short club name). |
| destination_team_id | character | Club id the player joined; null when the destination is not a league club. |
| destination_team_name | character | Name of the club the player joined (the free-text field, else the short club name). |

## See Also

Other SCB Functions: `scb_game()`, `scb_game_actions()`, `scb_game_players()`, `scb_game_teams()`, `scb_games()`, `scb_games_recent()`, `scb_player_statistics()`, `scb_playoff_games()`, `scb_playoff_series()`, `scb_records_per_game()`, `scb_records_season_averages()`, `scb_referees()`, `scb_season_player()`, `scb_season_players()`, `scb_season_teams()`, `scb_seasons()`, `scb_standings()`, `scb_team()`, `scb_team_rosters()`, `scb_team_seasons()`, `scb_team_statistics()`, `scb_transfers_teams()`

`scb_transfers_teams` **Get BBL Team Transfers**

## Description

Per club, the players joining and leaving, one row per move with its direction (BBL only).

Endpoint: `GET https://api.basketball-bundesliga.de/transfers/teams`

## Usage

```
scb_transfers_teams()
```

## Details

- Unofficial use of a public source: these wrappers read the front-end API of the German Bundesliga sites (easycredit-bbl.de, toyota-dbbl.de), run by SCB; payloads are not redistributed as release assets.
- Legal notice: the BBL Impressum forbids storing BBL content (`Inhalte`) in an information system without BBL's written permission. That is a copyright term, not a robots rule (the API host has no robots.txt). The owner reviewed it and accepted the intake (2026-10-12 ruling); keep requests slow (the runtime waits at least 1 s per host) and raise any published data release with the owner. The DBBL host is the same operator and API; its site notice was not separately reviewed.
- Both hosts need two headers: a constant `x-api-key` and an `x-api-secret` that rotates. The runtime derives the secret on first use from the key embedded in an easycredit-bbl.de page (reversed, `:` turned into `-`) and never logs or returns it. The derivation was reverse-engineered from observed requests and responses and has not been checked against the site's own front-end code; if the site changes how it signs requests, every call fails with `wehoop_asset_fetch`.
- Id key spaces: seasonId is the START year (2025 = 2025-26), so a wrapper's END-year `season` is sent as `season - 1`. Teams are keyed by teamId (seasonTeamId equals it in every sample). seasonPlayerId is per player per season and per club (a player traded mid-season has two); realPlayerId is the person. playerId is ambiguous (the seasonPlayerId in box scores, records and /season-players; the person in /transfers). Game ids are strings.
- Errors are classed conditions: `wehoop_no_data` (404, 204, a 400 "not a valid" id, no rows, a box score of a game not yet started), `wehoop_bad_request` (a bad argument, another 400 / 422) and `wehoop_asset_fetch` (a refused secret after one re-derive, any other non-2xx, an HTML / non-JSON / error-shaped body, a truncated page, a body for another game, season, club or league, a box score still in play), all inheriting `wehoop_error`. Wrappers raise; they never return an empty frame.

```
scb_transfers_teams()
```

## Value

A wehoop_data tibble: Per club, the players joining and leaving, one row per move with its direction (BBL only). An empty valid answer raises wehoop_no_data; the columns are always these, in this order (sdv-py scb_transfers_teams):

| col_name | type | description |
|---|---|---|
| team_id | character | Club id, a string; equal to season_team_id in every sample. Joins to scb_team and scl |
| direction | character | joining or leaving: whether team_id is the destination or the source of the move. |
| league | character | League the row belongs to: bbl (men) or dbbl (women); BBL and DBBL game ids ove |
| transfer_date | character | Date of the transfer, YYYY-MM-DD (the API's sortingDate). |
| season_player_id | character | Season-player id: one player's engagement with one club (kept across seasons while he |
| real_player_id | integer | Global person id (integer): the same player across clubs and seasons. Join players acros |
| first_name | character | Player's given name. |
| last_name | character | Player's family name. |
| position | character | Position: POINT_GUARD, SHOOTING_GUARD, SMALL_FORWARD, POWER_FC |
| source_team_id | character | Club id the player left; null when the source is not a league club. |
| source_team_name | character | Name of the club the player left (the free-text field, else the short club name). |
| destination_team_id | character | Club id the player joined; null when the destination is not a league club. |
| destination_team_name | character | Name of the club the player joined (the free-text field, else the short club name). |

## See Also

Other SCB Functions: scb_game(), scb_game_actions(), scb_game_players(), scb_game_teams(), scb_games(), scb_games_recent(), scb_player_statistics(), scb_playoff_games(), scb_playoff_series(), scb_records_per_game(), scb_records_season_averages(), scb_referees(), scb_season_player(), scb_season_players(), scb_season_teams(), scb_seasons(), scb_standings(), scb_team(), scb_team_rosters(), scb_team_seasons(), scb_team_statistics(), scb_transfers()

---

# tbf_daily_matches TBF Daily Matches

---

## Description

Every TBF game played on one date, grouped by competition, optionally for one league-season.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/get-daily-matches

## Usage

```
tbf_daily_matches(match_date, season = NULL, competition = NULL)
```

## Arguments

| | |
|---|---|
| match_date | Required. Game date 'YYYY-MM-DD' (Turkey local date), e.g. '2026-10-11'. |

| | |
|---|---|
| season | Optional, with competition: Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. Default: games of every competition. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Optional, with season (a season alone reads the kbsl list). Default: every competition. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. One row per game with its group_name (the competition group the app shows). Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| group_name | character | Competition group the app shows the game under (grup adi). |
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons |
| match_code | character | TBF short game code ('1EA4917'); unique per game, meaning of the letters unverified. |
| genius_id | integer | Genius Sports LiveStats match id (TBF spells it genuisId); joins the genius family. Null |
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: B |
| activity_name | character | Activity name ('BSL 2026-2027'). |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league |

| | | |
|---|---|---|
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-202 |
| season_label | character | Season label as TBF writes it, 'YYYY-YYYY' (start and end year). |
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label |
| tipoff | POSIXct | Tip-off as a naive timestamp in Turkey local time (Europe/Istanbul, UTC+3); the API se |
| match_date | Date | Tip-off date 'YYYY-MM-DD'; equals the MatchDate parameter (checked). |
| week | character | Week (hafta) label or number: 'N. Hafta' on schedule rows, the number on standings ro |
| phase | integer | Phase code as a string ('1' = first half of the regular season); see get-league-weeks devre |
| status_id | integer | Game status code: 1 = played (Oynandi), 4 = not played (Oynanmadi), 19 = fixture with |
| home_team_process_id | integer | Team-season id (teamProcessId = takimIslemId, 'team registration'; key space: TBF tea |
| home_team | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: |
| home_score | integer | Home points (null before the game). |
| away_team_process_id | integer | Team-season id (teamProcessId = takimIslemId, 'team registration'; key space: TBF tea |
| away_team | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: |
| away_score | integer | Away points (null before the game). |
| venue | character | Venue (salon) name. |
| province | character | Province (il) of the venue when sent, uppercase Turkish; null otherwise. |
| gender | character | Gender in Turkish when sent; empty on get-daily-matches, so null. |
| broadcast_channel | character | Broadcaster ('beINSports', 'HT SPOR', 'Youtube / TBF TV'), or ''. |
| has_overtime | logical | Overtime flag (uzatma = overtime). Reliability unverified: false on every one of 64 BSL |
| status | character | Game status text in Turkish: Oynandi = played, Oynanmadi = not played. |

## See Also

Other TBF Functions: `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_daily_matches(match_date = "2026-10-11"))
```

---

`tbf_game_header` **TBF Game Header**

---

## Description

Game header of a finished game: teams, coaches, final score, officials, venue and the Genius LiveStats id.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-header`

## Usage

```
tbf_game_header(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasor |
| match_code | character | TBF short game code ('1EA4917'). |
| genius_id | integer | Genius Sports LiveStats match id (TBF spells it genuisId); joins the genius family. Nu |
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: |
| activity_name | character | Activity name ('BSL 2026-2027'). |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2( |
| season_label | character | Season label 'YYYY-YYYY' (seasonName). |
| season | integer | Season END year, read from the season label (seasonName). |
| tipoff | POSIXct | Tip-off as a naive timestamp in Turkey local time (Europe/Istanbul, UTC+3). |

| | | |
|---|---|---|
| status_id | integer | Game status code: 1 = played (Oynandi), 4 = not played (Oynanmadi), 19 = fixture wi |
| status | character | Game status text in Turkish: Oynandi = played. Only finished games are served. |
| venue | character | Venue name (location) as TBF prints it. |
| attendance | integer | Spectator count (matchSpectatorCount); 0 when not recorded. |
| has_overtime | logical | Overtime flag (uzatma = overtime). Reliability unverified: false on every one of 64 BS |
| broadcast_channel | character | Broadcaster (matchBroadcastChannel), e.g. beINSports, or null. |
| home_team_process_id | integer | Home team-season id (key space: TBF team registrations; the header calls it homeTea |
| home_team | character | Home team name as registered for that season, uppercase with Turkish letters. |
| home_coach | character | Home head coach name, uppercase. |
| home_score | integer | Home final score. |
| away_team_process_id | integer | Away team-season id (key space: TBF team registrations). |
| away_team | character | Away team name as registered for that season, uppercase with Turkish letters. |
| away_coach | character | Away head coach name, uppercase. |
| away_score | integer | Away final score. |
| referee | character | Main referee's name (names only: no contact or birth data is returned). |
| first_assistant_referee | character | First assistant referee's name. |
| second_assistant_referee | character | Second assistant referee's name. |
| commissioner | character | Commissioner's (temsilci = representative) name. |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

## Examples

```
try(tbf_game_header(match_id = 346279))
```

---

# tbf_game_leaders TBF Game Leaders

---

## Description

Game leaders of a finished game per statistic category (top five each).

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-leader`

## Usage

```
tbf_game_leaders(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_game_leaders(match_id = 346279)
```

## Value

A wehoop_data tibble. One row per leader, best first within a category. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62552 |
| category | character | Leader category code (leaderCategory), e.g. VerimlilikPuani = efficiency rating. |
| category_label | character | Leader category label in Turkish (leaderCategoryDescription), e.g. Verimlilik Puani. |
| rank | integer | Position in the category's list, 1 = best (top five per category). |
| side | character | Side of the leader's team (isHome). |
| player_id | integer | TBF person id of the leader (key space: TBF persons; equals the box score player_id). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| value | double | The leader's value in the category (points, rebounds, ...). |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

# tbf_game_players TBF Game Players

---

## Description

Game roster from the header: every listed player of both sides (names and shirt numbers, no usable ids).

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-header`

## Usage

```
tbf_game_players(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked

for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_game_players(match_id = 346279)
```

## Value

A `wehoop_data` tibble. One row per rostered player. The header carries no usable player id; join on name + shirt number + side. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62552 |
| side | character | Side of the game the row belongs to: home or away (from the route's isHome flag or home / awa |
| team | character | Team name for the roster row; empty in the samples, so null. |
| player_name | character | Player name as the header roster lists it. |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| is_starter | logical | True for a starter in the header roster (isStarter). |
| is_active | logical | True when the player is flagged active in the header (isActive). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

# `tbf_highlights` TBF Highlights

## Description

Team comparison of a finished game: rebounds, paint points, fast-break points, lead changes and more.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-ozet`

## Usage

```
tbf_highlights(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_highlights(match_id = 346279)
```

## Value

A wehoop_data tibble. One row per statistic with the home and away value. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62552 i |
| title | character | Statistic name in Turkish (Ribaund = rebounds, Asist = assists, Top Kaybi = turnovers). |
| category | character | Statistic code (leaderCategory), e.g. PointsFromPaint, LeadChanges, TimeLeading. |
| home_value | character | Home value exactly as sent (a string: '90', or '28:38' for TimeLeading). |
| away_value | character | Away value exactly as sent (a string: '89', or '8:19' for TimeLeading). |
| home_number | double | Home value as a number; null for a time value such as TimeLeading. |
| away_number | double | Away value as a number; null for a time value such as TimeLeading. |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

# tbf_leaders_assists **TBF Leaders: Assists**

---

## Description

Season leaders in assists: the server's top 20 by season total and by per-game average.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-asist-leaders

## Usage

```
tbf_leaders_assists(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth

dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_leaders_assists(competition = "bsl", season = 2027)
```

## Value

A `wehoop_data` tibble. Up to 20 rows per `kind` (`total` and `average`), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

`tbf_leaders_blocks` **TBF Leaders: Blocks**

## Description

Season leaders in blocks: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-blok-leaders`

## Usage

```
tbf_leaders_blocks(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_leaders_blocks(competition = "bsl", season = 2027)
```

## Value

A wehoop_data tibble. Up to 20 rows per kind (total and average), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

tbf_leaders_efficiency

# TBF Leaders: Efficiency

---

## Description

Season leaders in efficiency rating: the server's top 20 by season total and by per-game average.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-mvp-leaders

## Usage

```
tbf_leaders_efficiency(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_leaders_efficiency(competition = "bsl", season = 2027)
```

## Value

A wehoop_data tibble. Up to 20 rows per kind (total and average), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |

| | | |
|---|---|---|
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

# `tbf_leaders_minutes` TBF Leaders: Minutes

---

## Description

Season leaders in minutes played: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-duration-leaders`

## Usage

```
tbf_leaders_minutes(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as

release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_leaders_minutes(competition = "bsl", season = 2027)
```

## Value

A `wehoop_data` tibble. Up to 20 rows per `kind` (`total` and `average`), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`,

tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

`tbf_leaders_points` **TBF Leaders: Points**

---

## Description

Season leaders in points: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-point-leaders`

## Usage

```
tbf_leaders_points(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished

game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_leaders_points(competition = "bsl", season = 2027)
```

## Value

A `wehoop_data` tibble. Up to 20 rows per `kind` (`total` and `average`), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

# `tbf_leaders_rebounds` TBF Leaders: Rebounds

## Description

Season leaders in rebounds: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-ribound-leaders`

## Usage

```
tbf_leaders_rebounds(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_leaders_rebounds(competition = "bsl", season = 2027)
```

## Value

A wehoop_data tibble. Up to 20 rows per kind (total and average), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |

| | | |
|---|---|---|
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

# `tbf_leaders_steals` TBF Leaders: Steals

---

## Description

Season leaders in steals: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-top-calma-leaders`

## Usage

```
tbf_leaders_steals(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as

release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_leaders_steals(competition = "bsl", season = 2027)
```

## Value

A `wehoop_data` tibble. Up to 20 rows per `kind` (`total` and `average`), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`,

tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

`tbf_leaders_threes` **TBF Leaders: Threes**

---

## Description

Season leaders in three-point field goals made: the server's top 20 by season total and by per-game average.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-3Point-leaders`

## Usage

```
tbf_leaders_threes(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422;

wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_leaders_threes(competition = "bsl", season = 2027)
```

## Value

A wehoop_data tibble. Up to 20 rows per kind (total and average), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

# tbf_leaders_turnovers **TBF Leaders: Turnovers**

---

## Description

Season leaders in turnovers: the server's top 20 by season total and by per-game average.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-top-kaybi-leaders

## Usage

```
tbf_leaders_turnovers(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_leaders_turnovers(competition = "bsl", season = 2027)
```

## Value

A wehoop_data tibble. Up to 20 rows per kind (total and average), best first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the r |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| category | character | Statistic of the wrapper: points, assists, blocks, rebounds, steals, minutes, threes, turnovers or e |
| kind | character | total = the server's top 20 by season TOTAL; average = the top 20 by PER-GAME average. |

| | | |
|---|---|---|
| rank | integer | Rank in the list, 1 = best (the API's order). |
| person_id | integer | TBF person id (oyuncuKodu, sent as a string; key space: TBF persons). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| team_process_id | integer | Team-season id (takimKodu, sent as a string; key space: TBF team registrations; despite the na |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'PIZZ |
| games | integer | Games played (macSayisi, sent as a string). |
| value | double | The ranked statistic: the season total for kind total, the per-game average for kind average (the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

`tbf_league_seasons` **TBF League Seasons**

---

## Description

Every season of one TBF league (bsl back to 1966-67) with its activity and season ids.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/League/get-leagues-and-seasons-by-pref`

## Usage

```
tbf_league_seasons(competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per season of the league, newest first. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: BSL 2026- |
| activity_name | character | League-season name as TBF prints it ('BSL 2026-2027'; the top flight reads 'TBL 2006-2007' u |
| url | character | League-season slug, '<prefix>-<YYYY>-<YYYY>' (bsl-2026-2027). |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2027, 172 = |
| season_label | character | Season label as TBF writes it, 'YYYY-YYYY' (start and end year). |
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the req |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix the |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_league_seasons(competition = "bsl"))
```

`tbf_leagues` **TBF Leagues**

## Description

The featured leagues (league-season activities) of one TBF season: BSL, TBL, KBSL and the others.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/League/get-leagues-for-filter`

## Usage

```
tbf_leagues(season)
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A wehoop_data tibble. One row per league: BSL, TBL, KBSL, TKBL, TB2L and the youth leagues with their activity_id. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: BSL 2026- |
| name | character | Activity name '<LEAGUE> YYYY-YYYY' ('BSL 2026-2027'). |
| display_name | character | League short name ('BSL', 'KBSL', 'TBL', 'TKBL', 'TB2L', 'BGL', 'BGLK', 'KBBL'). |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix the |
| gender | character | Gender in Turkish: 'Erkek' = men, 'Kadin' = women. |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2027, 172 = |
| season_label | character | Season label as TBF writes it, 'YYYY-YYYY' (start and end year). |
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the req |
| sort | integer | Display order in the app (integer, 1 = first: BSL). |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

## Examples

```
try(tbf_leagues(season = 2027))
```

---

# tbf_pbp TBF Play-by-Play

---

## Description

Play-by-play of a finished game (2025-26 onward), in game order, with the running score and an English gloss.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/game-flow

## Usage

```
tbf_pbp(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. One row per event. Games before 2025-26 raise wehoop_no_data. The feed has no player id. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62 |
| action_number | integer | Play sequence number (actionNumber); gaps are deleted or hidden actions. Rows are in the |
| period | integer | Period number (quarterNumber): 1-4; overtime numbering is unverified. |
| period_type | integer | Period type code: 1 = regulation, 2 = overtime (from the app code; no overtime game in the |
| clock | character | Game clock REMAINING in the period as 'mm:ss' (counts down from 10:00). |
| seconds_remaining | integer | Game clock remaining in the period, in seconds. |
| home_score | integer | Running home score after the play (the home half of the 'home-away' score text). |
| away_score | integer | Running away score after the play (the away half of the score text). |
| side | character | Side of the acting team (isHome); null on game events (period start / end, game end). |
| player_name | character | Acting player's name; null on game events. The feed carries no player id. |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| action | character | Play type in Turkish exactly as TBF sends it (77 values seen). |

| | | |
|---|---|---|
| action_en | character | English gloss of the Turkish action, from the refs action vocabulary; null for an action not in |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

## Examples

```
try(tbf_pbp(match_id = 346279))
```

---

# tbf_period_scores TBF Period Scores

---

## Description

Points each side scored in each period of a finished game, overtime included.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-ozet

## Usage

```
tbf_period_scores(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07

(no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per side and period (home first). Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62552 in 2 |
| side | character | Side of the score: home (homeTeamScore) or away (awayTeamScore). |
| period | integer | Period number (quarter): 1-4 regulation; overtime numbering is unverified. |
| period_type | integer | Period type code: 1 = regulation, 2 = overtime (from the app code; no overtime game in the sample |
| points | integer | Points the side scored IN the period (not cumulative). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_period_scores(match_id = 346279))
```

---

`tbf_player_box` **TBF Player Box Score**

---

## Description

Player box score of a finished game: minutes, shooting splits, rebounds, assists, +/- and efficiency.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-istatistik`

## Usage

```
tbf_player_box(match_id)
```

## Arguments

| | |
|---|---|
| `match_id` | Required. TBF game id (integer), e.g. 346279; match_id in `tbf_schedule()`. One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per player who is on the sheet; `is_starter` marks the starting five (none flagged in 2006-07). Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 6255 |
| side | character | Side of the game the row belongs to: home or away (from the route's isHome flag or home / av |
| team_process_id | integer | Team-season id of the player's side (from the game header; key space: TBF team registrations) |

| | | |
|---|---|---|
| player_id | integer | TBF person id (key space: TBF persons, stable across seasons and clubs: 196413). The same v |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| is_starter | logical | True for the starting five (homeTopFive / awayTopFive block); none are flagged in 2006-07. |
| seconds_played | integer | Time on court in seconds, parsed from the 'mm:ss' or 'hh:mm:ss' text the API sends (team tot |
| points | integer | Points scored in the game (or season) the row covers. |
| fg_made | integer | Field goals made, 2PT + 3PT (successfulShots). |
| fg_attempted | integer | Field goals attempted, 2PT + 3PT (shotAttempts). |
| fg_pct | integer | Field-goal percentage 0-100, rounded to an integer by the API. |
| fg2_made | integer | Two-point field goals MADE (the API calls it twoPointShots; attempts are fg2_attempted). |
| fg2_attempted | integer | Two-point field goals attempted (twoPointShotAttempts). |
| fg2_pct | integer | Two-point percentage 0-100, an integer. |
| fg3_made | integer | Three-point field goals MADE (the API calls it threePointShots). |
| fg3_attempted | integer | Three-point field goals attempted (threePointShotAttempts). |
| fg3_pct | integer | Three-point percentage 0-100, an integer. |
| ft_made | integer | Free throws MADE (serbest atis; the API calls it freeThrows). |
| ft_attempted | integer | Free throws attempted (freeThrowAttempts). |
| ft_pct | integer | Free-throw percentage 0-100, an integer. |
| off_rebounds | integer | Offensive rebounds (hucum ribaundu). |
| def_rebounds | integer | Defensive rebounds (savunma ribaundu). |
| rebounds | integer | Total rebounds (toplam ribaund). |
| assists | integer | Assists in this game (asist). |
| steals | integer | Steals in this game (top calma). |
| blocks | integer | Shots this player blocked in this game (blok). |
| turnovers | integer | Turnovers committed in this game (top kaybi). |
| fouls | integer | Personal fouls committed (faul). |
| fouls_drawn | integer | Fouls drawn (rivalFoul = foul committed by the opponent). |
| blocks_received | integer | Own shots blocked by an opponent (blocksReceived). |
| plus_minus | integer | Plus/minus while on court (plusMinus); null when the API sends ". |
| efficiency | integer | Efficiency rating (verimlilik puani, PIR): points + rebounds + assists + steals + blocks + fouls |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_player_box(match_id = 346279))
```

`tbf_player_career` **TBF Player Career**

## Description

One player's career: a row per season and competition with totals (birth date is not returned).

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-player-detail-profile-and-c`

## Usage

```
tbf_player_career(player_id)
```

## Arguments

| | |
|---|---|
| `player_id` | Required. TBF person id, e.g. 196413; person_id in `tbf_players()` and the box score. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

```
tbf_player_career(player_id = 196413)
```

## Value

A `wehoop_data` tibble. One row per career season and competition. Totals and averages rows are not included. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| person_id | integer | TBF person id of the player asked for (the player_id argument). |
| player_name | character | Player's full name from the profile (the same on every row). |
| nationality | character | Nationality as a TURKISH country name from the profile (country). |
| height_cm | integer | Height in centimetres from the profile; null when TBF has none. |
| season_label | character | Career season as TBF writes it (seasonCode), e.g. 2025-2026. |
| season | integer | Career season END year, read from the season label. |
| team_name | character | Team the player represented that season, uppercase with Turkish letters (teamName). |
| activity_name | character | League-season name (activityName), e.g. 'BSL 2025-2026'. |
| activity_type_id | integer | Competition type id (faaliyetTuruId): 1 = BSL, 4 = TBL, 79 = BGL. |
| activity_id | integer | Activity id of the league-season (faaliyetAltAlaniId; key space: TBF league-seasons). |
| games | integer | Games played (macSayisi = game count). |
| seconds_played | integer | Total time on court that season in seconds (formattedTotalSeconds 'hh:mm:ss'). |
| points | integer | Points scored in the game (or season) the row covers. |
| assists | integer | Assists in that season and competition. |
| fg_made | integer | Field goals made (fieldGoalsMade), 2PT + 3PT. |
| fg_attempted | integer | Field goals attempted (fieldGoalsAttempted), 2PT + 3PT. |
| fg3_made | integer | Three-point field goals made (threePointsMade). |
| fg3_attempted | integer | Three-point field goals attempted (threePointsAttempted). |
| ft_made | integer | Free throws made (the API spells the field freeThows). |
| ft_attempted | integer | Free throws attempted (freeThowsAttempted). |
| off_rebounds | integer | Offensive rebounds in that season and competition. |
| def_rebounds | integer | Defensive rebounds in that season and competition. |
| rebounds | integer | Total rebounds in that season and competition. |
| steals | integer | Steals in that season and competition. |
| turnovers | integer | Turnovers (the API spells the field tournovers). |
| blocks | integer | Blocks in that season and competition. |
| fouls | integer | Personal fouls (the API spells the field fauls). |
| efficiency | integer | Efficiency rating total (playerEfficiency, PIR) for that season and competition. |
| fouls_drawn | integer | Fouls drawn (yapilanFaul). |
| blocks_received | integer | Own shots blocked by an opponent (yenilenBlok). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

`tbf_players` **TBF Players**

## Description

Every player registered in one league-season (birth dates are not returned).

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Player/get-player-list`

## Usage

```
tbf_players(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per registered player. Birth dates are not returned. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year of the request. |
| competition | character | League prefix the row reports (bsl, tbl or kbsl); equals the request's competition. |
| person_id | integer | TBF person id (kisiID; key space: TBF persons, stable across seasons and clubs). |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| nationality | character | Nationality as a TURKISH country name (ABD = United States); see the nationality_code. |
| nationality_code | character | Nationality code (tabiyetKisa), ISO 3166-1 alpha-3 except GB for the United Kingdom. |
| club_id | integer | Stable club id (takimID / teamId of the team lists; key space: TBF clubs, the same across seasc |
| team_name | character | Team name for that season, uppercase with Turkish letters. |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| prefix | character | League prefix of the row (bsl, tbl, kbsl); the runtime refuses any other than the request's. |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2027, 172 |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_players(competition = "bsl", season = 2027))
```

---

`tbf_schedule` **TBF Schedule**

---

## Description

Schedule and results of one league-season: the games of one week, or of every week.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/get-all-matches-for-filter`

## Usage

```
tbf_schedule(season, competition = "kbsl", week = NULL)
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |
| week | Week (hafta) number as returned by tbf_weeks(), e.g. 1; playoff rounds continue the numbering (31-33 in 2025-26). Default: the latest week for standings, every week for the schedule. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. One row per game. Scores are null before the game; status_id 1 = played, 4 = not played, 19 = no date yet. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons |
| match_code | character | TBF short game code ('1EA4917'); unique per game, meaning of the letters unverified. |
| genius_id | integer | Genius Sports LiveStats match id (TBF spells it genuisId); joins the genius family. Null |

| | | |
|---|---|---|
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: B |
| activity_name | character | Activity name ('BSL 2026-2027'). |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-202 |
| season_label | character | Season label as TBF writes it, 'YYYY-YYYY' (start and end year). |
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label |
| tipoff | POSIXct | Tip-off as a naive timestamp in Turkey local time (Europe/Istanbul, UTC+3); the API se |
| match_date | Date | Tip-off date 'YYYY-MM-DD' (Turkey local). |
| week | character | Week label: 'N. Hafta' (week N) or the playoff round ('Playoff Final'). |
| phase | integer | Phase code as an integer: 1 = first half of the regular season; see tbf_weeks. |
| status_id | integer | Game status code: 1 = played (Oynandi), 4 = not played (Oynanmadi), 19 = fixture with |
| home_team_process_id | integer | Team-season id (teamProcessId = takimIslemId, 'team registration'; key space: TBF tea |
| home_team | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: |
| home_score | integer | Home points (null before the game; the API sends '' ). |
| away_team_process_id | integer | Team-season id (teamProcessId = takimIslemId, 'team registration'; key space: TBF tea |
| away_team | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: |
| away_score | integer | Away points (null before the game; the API sends '' ). |
| venue | character | Venue (salon) name. |
| province | character | Province (il) of the venue, uppercase Turkish ('ISTANBUL'). |
| gender | character | Gender in Turkish: Erkek = men, Kadin = women. |
| broadcast_channel | character | Broadcaster ('beINSports', 'HT SPOR', 'Youtube / TBF TV'), or ''. |
| has_overtime | logical | Overtime flag (uzatma = overtime). Reliability unverified: false on every one of 64 BSL |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_schedule(competition = "bsl", season = 2027, week = 1))
```

---

`tbf_seasons` **TBF Seasons**

---

## Description

The 20 most recent TBF seasons, domestic and international (a server cap), newest first.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Season/get-seasons`

## Usage

```
tbf_seasons()
```

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per season, newest first. `season` is the END year of the label (2026-2027 -> 2027); `international` marks the odd `UA-` seasons. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2027, 172 = 2 |
| season_label | character | Season label as TBF writes it, 'YYYY-YYYY' (start and end year). |
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the requ |
| start_date | Date | Season start, Turkey local time (Europe/Istanbul, UTC+3), ISO 8601 without zone: domestic seasc |
| end_date | Date | Season end, Turkey local time (Europe/Istanbul, UTC+3), ISO 8601 without zone ('2027-06-30T2 |
| international | logical | True for the international UA- seasons (odd season ids); false for domestic ones. |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`,

tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

## Examples

```
try(tbf_seasons())
```

---

`tbf_shots` **TBF Shots**

---

## Description

Every field-goal attempt of a finished game (2025-26 onward) with its court position.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/shot-chart`

## Usage

```
tbf_shots(match_id)
```

## Arguments

| | |
|---|---|
| `match_id` | Required. TBF game id (integer), e.g. 346279; match_id in `tbf_schedule()`. One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked

for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per attempt. `x` and `y` are percentages of a horizontal full court. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62552 |
| period | integer | Period number of the attempt: 1-4; overtime numbering is unverified. |
| period_type | integer | Period type code: 1 = regulation, 2 = overtime (from the app code; no overtime game in the sam |
| side | character | Side of the shooting team (isHome). |
| shot_player_id | integer | Shooter id in the SHOT-CHART key space, which is NOT the TBF person id of the box score (Q |
| player_name | character | Player name as TBF prints it for that route (title case, Turkish letters kept). |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| x | integer | Court x position: integer percent (0-100) of the LENGTH of a horizontal full court from the LE |
| y | integer | Court y position: integer percent (0-100) of the court WIDTH from the BOTTOM sideline (bask |
| made | logical | True when the attempt was made (isSucceed). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_shots(match_id = 346279))
```

---

`tbf_standings` **TBF Standings**

---

## Description

League table of one league-season after a week (default the latest), one row per team and group.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/League/get-standings-table`

## Usage

```
tbf_standings(season, competition = "kbsl", week = NULL)
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |
| week | Week (hafta) number as returned by tbf_weeks(), e.g. 1; playoff rounds continue the numbering (31-33 in 2025-26). Default: the latest week for standings, every week for the schedule. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. One row per team and group. Tie-break detail rows (tiedRows) are not included. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league pr |
| group | character | Group name (grup); empty for a single-group league. |
| activity_id | integer | Activity id (faaliyet = activity; key space: TBF league-SEASONS, new every season: BSI |
| team_process_id | integer | Team-season id (teamProcessId = takimIslemId, a team REGISTRATION; key space: TB |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters (sponsor name: 'P |
| week | integer | The week the table is computed after (hafta); equals the week argument, else the latest. |
| rank | integer | League rank (sira) after that week. |
| played | integer | Games played (o = oynanan). |
| won | integer | Games won (g = galibiyet). |
| lost | integer | Games lost (m = maglubiyet). |
| points_for | integer | Points scored (a = atilan). |
| points_against | integer | Points conceded (y = yenilen). |
| points | integer | League points (puan): 2 for a win, 1 for a loss. |
| head_to_head_points | integer | Head-to-head league points among teams tied on points (icP = ic puan). |
| head_to_head_diff | integer | Head-to-head point difference among teams tied on points (icA = ic averaj). |
| total_diff | integer | Total point difference over all games (toA = toplam averaj). |
| excluded | integer | Exclusion marker (ihrac = expulsion); 0 for teams still in the league. |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_standings(competition = "bsl", season = 2027))
```

---

`tbf_team_box` **TBF Team Box Score**

---

## Description

Team totals of a finished game with the team-credited rebounds, turnovers and fouls.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Match/mac-istatistik`

## Usage

```
tbf_team_box(match_id)
```

## Arguments

| | |
|---|---|
| match_id | Required. TBF game id (integer), e.g. 346279; match_id in tbf_schedule(). One global sequence across leagues and seasons. Only a finished game is served. |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. Two rows: home, then away. Minutes are the players' summed seconds. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| match_id | integer | TBF game id (key space: TBF games, ONE global sequence across leagues and seasons: 62 |
| side | character | Side of the game the row belongs to: home or away (from the route's isHome flag or home / |
| team_process_id | integer | Team-season id of the side (from the game header; key space: TBF team registrations). |
| seconds_played | integer | Time on court in seconds, parsed from the 'mm:ss' or 'hh:mm:ss' text the API sends (team |
| points | integer | Points scored in the game (or season) the row covers. |
| fg_made | integer | Team field goals made, 2PT + 3PT. |
| fg_attempted | integer | Team field goals attempted, 2PT + 3PT. |
| fg_pct | integer | Team field-goal percentage 0-100, rounded to an integer by the API. |
| fg2_made | integer | Team two-point field goals MADE (the API calls it twoPointShots). |
| fg2_attempted | integer | Team two-point field goals attempted. |
| fg2_pct | integer | Team two-point percentage 0-100, an integer. |
| fg3_made | integer | Team three-point field goals MADE (the API calls it threePointShots). |

| | | |
|---|---|---|
| fg3_attempted | integer | Team three-point field goals attempted. |
| fg3_pct | integer | Team three-point percentage 0-100, an integer. |
| ft_made | integer | Team free throws MADE. |
| ft_attempted | integer | Team free throws attempted. |
| ft_pct | integer | Team free-throw percentage 0-100, an integer. |
| off_rebounds | integer | Offensive rebounds (hucum ribaundu). |
| def_rebounds | integer | Defensive rebounds (savunma ribaundu). |
| rebounds | integer | Total rebounds (toplam ribaund). |
| assists | integer | Assists in this game (asist). |
| steals | integer | Steals in this game (top calma). |
| blocks | integer | Shots this player blocked in this game (blok). |
| turnovers | integer | Turnovers committed in this game (top kaybi). |
| fouls | integer | Personal fouls committed (faul). |
| fouls_drawn | integer | Fouls drawn by the team (rivalFoul). |
| blocks_received | integer | Team shots blocked by the opponent. |
| efficiency | integer | Team efficiency rating (verimlilik puani): the players' ratings summed; a number on team r |
| team_off_rebounds | integer | Team offensive rebounds not credited to a player (homeTeamStats). |
| team_def_rebounds | integer | Team defensive rebounds not credited to a player. |
| team_rebounds | integer | Team rebounds not credited to a player (offensive + defensive). |
| team_turnovers | integer | Team turnovers not credited to a player. |
| team_fouls | integer | Team fouls not credited to a player (bench / coach technicals); 0 in the samples. |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_team_box(match_id = 346279))
```

---

`tbf_team_games` **TBF Team Games**

---

## Description

Games and results of one team-season, scheduled games included.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Team/get-team-detail-matches-by-season`

## Usage

```
tbf_team_games(season, team_process_id, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| team_process_id | Required. Team-season id (team registration), e.g. 265995 = Tofas in BSL 2026-27; team_process_id in tbf_teams() and tbf_schedule(). New for every club in every league-season; not the stable club_id. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_team_games(season = 2027, team_process_id = 265995)
```

## Value

A wehoop_data tibble. One row per game of the team (scheduled games included). Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in

it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league |
| team_process_id | integer | Team-season id of the request (key space: TBF team registrations). |
| match_id | integer | TBF game id (key space: TBF games, one global sequence). |
| match_code | character | TBF short game code. |
| tipoff | POSIXct | Tip-off as a naive timestamp in Turkey local time (tarih). |
| week | character | Week label (formattedWeek), e.g. '1. Hafta'. |
| city | character | City of the venue (sehir), uppercase Turkish. |
| venue | character | Venue name (salon). |
| home_team_process_id | integer | Home team-season id (takimAId; key space: TBF team registrations). |
| home_team | character | Home team name (takimA). |
| home_score | integer | Home points (skorA); null before the game. |
| away_team_process_id | integer | Away team-season id (takimBId). |
| away_team | character | Away team name (takimB). |
| away_score | integer | Away points (skorB); null before the game. |
| team_is_home | logical | True when the asked team was the home side (isHome). |
| team_won | logical | True when the asked team won (isWin); false for a loss or an unplayed game. |
| is_played | logical | True once the game has been played (isPlayed). |
| status | character | Game status text in Turkish: Oynandi = played, Oynanmadi = not played (durum). |
| status_id | integer | Game status code (durumId): 1 = played, 4 = not played. |
| has_overtime | logical | Overtime flag (uzatma = overtime). Reliability unverified: false on every one of 64 BSL |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_player_stats()`, `tbf_team_roster()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

---

# `tbf_team_player_stats` TBF Team Player Stats

---

## Description

Season player statistics of one team-season: totals, percentages and per-game averages.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Team/get-team-detail-statistic`

## Usage

```
tbf_team_player_stats(season, team_process_id, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| team_process_id | Required. Team-season id (team registration), e.g. 265995 = Tofas in BSL 2026-27; team_process_id in tbf_teams() and tbf_schedule(). New for every club in every league-season; not the stable club_id. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_team_player_stats(season = 2027, team_process_id = 265995)
```

## Value

A wehoop_data tibble. One row per player. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|

| | | |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league pr |
| team_process_id | integer | Team-season id of the request (key space: TBF team registrations). |
| person_id | integer | TBF person id (code; key space: TBF persons, stable across seasons and clubs). |
| jersey_number | character | Shirt number (oyuncuNo) as a string. |
| first_name | character | Player's first name. |
| last_name | character | Player's last name. |
| games | integer | Games played (macSayisi). |
| seconds_played | integer | Season total time on court in seconds (sureToplam 'hh:mm:ss'). |
| points | integer | Season total points (sayiToplam; the API sends it as a string). |
| fg_made | integer | Season field goals made (agBasarili; sent as a string). |
| fg_attempted | integer | Season field goals attempted (agDeneme). |
| fg2_made | integer | Season two-point field goals made (puan20Toplam). |
| fg2_attempted | integer | Season two-point field goals attempted (puan21Toplam). |
| fg3_made | integer | Season three-point field goals made (puan30Toplam). |
| fg3_attempted | integer | Season three-point field goals attempted (puan31Toplam). |
| ft_made | integer | Season free throws made (sA0Toplam). |
| ft_attempted | integer | Season free throws attempted (sA1Toplam). |
| def_rebounds | integer | Season defensive rebounds (drToplam). |
| off_rebounds | integer | Season offensive rebounds (hrToplam). |
| rebounds | integer | Season total rebounds (trToplam). |
| assists | integer | Season assists (asistToplam). |
| blocks | integer | Season blocks (blokToplam). |
| steals | integer | Season steals (topCalmaToplam). |
| turnovers | integer | Season turnovers (topKaybiToplam). |
| fouls | integer | Season personal fouls (faulToplam). |
| plus_minus | double | Season plus/minus contribution (oyunaKatki = contribution to the game), a decimal string. |
| efficiency | integer | Season efficiency rating total (vpToplam, verimlilik puani). |
| fg_pct | double | Field-goal percentage 0-100 with two decimals (agYuzdesi). |
| fg2_pct | double | Two-point percentage 0-100 with two decimals (ikilikYuzdesi). |
| fg3_pct | double | Three-point percentage 0-100 with two decimals (uclukYuzdesi). |
| ft_pct | double | Free-throw percentage 0-100 with two decimals (serbestAtisYuzdesi). |
| seconds_per_game | integer | Average time on court per game in seconds (sureOrtalama 'hh:mm:ss'). |
| points_per_game | double | Points per game (sayiOrtalama). |
| rebounds_per_game | double | Rebounds per game (trOrtalama). |
| assists_per_game | double | Assists per game (asistOrtalama). |
| steals_per_game | double | Steals per game (topCalmaOrtalama). |
| blocks_per_game | double | Blocks per game (blokOrtalama). |
| turnovers_per_game | double | Turnovers per game (topKaybiOrtalama). |
| efficiency_per_game | double | Efficiency rating per game (vpOrtalama). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`,

tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_roster(), tbf_team_staff(), tbf_teams(), tbf_weeks()

---

tbf_team_roster **TBF Team Roster**

---

## Description

Roster of one team-season with games, minutes and per-game points and efficiency (birth dates dropped).

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Team/get-team-active-player-list-by-se

## Usage

```
tbf_team_roster(season, team_process_id, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| team_process_id | Required. Team-season id (team registration), e.g. 265995 = Tofas in BSL 2026-27; team_process_id in tbf_teams() and tbf_schedule(). New for every club in every league-season; not the stable club_id. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per rostered person. Birth dates are not returned. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league pr |
| team_process_id | integer | Team-season id of the request (key space: TBF team registrations). |
| person_id | integer | TBF person id (personId; key space: TBF persons). |
| first_name | character | Given name as registered with TBF, mixed case ('Yigitcan'). |
| last_name | character | Family name as registered with TBF, mixed case ('Saybir'). |
| nationality | character | Nationality as a TURKISH country name (country); there is no code on this route. |
| height_cm | integer | Height in centimetres; null when TBF has none. |
| jersey_number | character | Shirt number as a string, so '0' and '00' stay distinct. |
| games | integer | Games played (macSayisi = game count). |
| seconds_played | integer | Season time on court in seconds (sure 'hh:mm:ss'). |
| points_per_game | double | Points per game (sayi). |
| efficiency_per_game | double | Efficiency rating per game (verimlilikPuani). |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`, `tbf_schedule()`, `tbf_seasons()`, `tbf_shots()`, `tbf_standings()`, `tbf_team_box()`, `tbf_team_games()`, `tbf_team_player_stats()`, `tbf_team_staff()`, `tbf_teams()`, `tbf_weeks()`

## Examples

```
try(tbf_team_roster(season = 2027, team_process_id = 265995))
```

`tbf_team_staff` **TBF Team Staff**

## Description

Coaching and club staff of one team-season.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/Team/get-team-detail-team-staff-by-sea`

## Usage

```
tbf_team_staff(season, team_process_id, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `team_process_id` | Required. Team-season id (team registration), e.g. 265995 = Tofas in BSL 2026-27; team_process_id in `tbf_teams()` and `tbf_schedule()`. New for every club in every league-season; not the stable club_id. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422;

wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

```
tbf_team_staff(season = 2027, team_process_id = 265995)
```

## Value

A wehoop_data tibble. One row per staff member. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| team_process_id | integer | Team-season id of the request (key space: TBF team registrations). |
| person_id | integer | TBF person id of the staff member (code; staff are TBF persons too). |
| first_name | character | Given name(s) as registered with TBF, uppercase ('MEVLUT TULGA'). |
| last_name | character | Family name as registered with TBF, uppercase ('ONGOREN'). |
| role | character | Staff role in Turkish as sent (job), e.g. Antrenor = head coach. |
| role_code | integer | Staff sub-role code (subRolesCode). |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_teams(), tbf_weeks()

---

# tbf_teams TBF Teams

## Description

Every team registered in one league-season with its team-season id (team_process_id) and club id.

Endpoint: GET https://miniappapi.tbf.org.tr/webapi-service/api/Team/get-teams-by-leauge

## Usage

```
tbf_teams(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| season | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises wehoop_no_data. |
| competition | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default "kbsl" (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (bsl Basketbol Super Ligi, back to 1966-67 in the league list; tbl second tier; kbsl women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises wehoop_asset_fetch). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises wehoop_no_data. Every game carries a Genius LiveStats id (genius_id, joins the genius family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: wehoop_no_data when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); wehoop_bad_request for an invalid argument or HTTP 400 / 422; wehoop_asset_fetch when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit wehoop_error. Timestamps are Turkey local time (Europe/Istanbul).

## Value

A wehoop_data tibble. One row per team registration. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises wehoop_no_data, as in sdv-py.

| col_name | type | description |
|---|---|---|
| season | integer | Season END year (2027 = the 2026-2027 season), read from the 'YYYY-YYYY' label or the |
| competition | character | TBF league the row belongs to: bsl, tbl or kbsl (the request's competition, or the league prefix |
| team_process_id | integer | Team-season id (teamProcessId = takimIslemId, a team REGISTRATION; key space: TBF tea |
| club_id | integer | Stable club id (takimID / teamId of the team lists; key space: TBF clubs, the same across seaso |
| team_name | character | Team name as registered for that season, uppercase with Turkish letters. |
| city | character | Home city (teamCityName), uppercase Turkish. |

| | | |
|---|---|---|
| code | character | Team short code (teamCode). |
| season_id | integer | Internal TBF season id (key space: TBF seasons, global across leagues: 174 = 2026-2027, 172 |
| season_label | character | Season label 'YYYY-YYYY' of the league-season. |

## See Also

Other TBF Functions: tbf_daily_matches(), tbf_game_header(), tbf_game_leaders(), tbf_game_players(), tbf_highlights(), tbf_leaders_assists(), tbf_leaders_blocks(), tbf_leaders_efficiency(), tbf_leaders_minutes(), tbf_leaders_points(), tbf_leaders_rebounds(), tbf_leaders_steals(), tbf_leaders_threes(), tbf_leaders_turnovers(), tbf_league_seasons(), tbf_leagues(), tbf_pbp(), tbf_period_scores(), tbf_player_box(), tbf_player_career(), tbf_players(), tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_weeks()

## Examples

```
try(tbf_teams(competition = "bsl", season = 2027))
```

---

`tbf_weeks` **TBF Weeks**

---

## Description

The weeks (rounds) of one league-season, playoff rounds included.

Endpoint: `GET https://miniappapi.tbf.org.tr/webapi-service/api/League/get-league-weeks`

## Usage

```
tbf_weeks(season, competition = "kbsl")
```

## Arguments

| | |
|---|---|
| `season` | Required. Season END year (2027 = the 2026-2027 season). TBF keys a league-season by internal season and activity ids; the runtime reads them from the league's season list by the END year of the label. A year the list lacks raises `wehoop_no_data`. |
| `competition` | TBF league: bsl (Basketbol Super Ligi), tbl (second tier) or kbsl (women's Super Ligi). The top flight was named TBL until 2007-08 and is listed under bsl. Default `"kbsl"` (the women's Super Ligi). |

## Details

Official TBF app API, unofficial for SportsDataverse: the backend of the Turkish Basketball Federation's mobile app (miniappapi.tbf.org.tr), which serves BSL, TBL, KBSL and the other TBF leagues. It needs no key; robots.txt answers 404 on the host, and the runtime spaces requests at least 1 s apart. SportsDataverse has not ruled on TBF's terms, so payloads are not redistributed as release assets. The vendor's origin hosts (nikayazilim.com) serve the same data without Cloudflare but are deliberately not used, redirects are never followed, and www.tbf.org.tr's interactive Turnstile is neither used nor automated.

Coverage: season and competition are parameters (`bsl` Basketbol Super Ligi, back to 1966-67 in the league list; `tbl` second tier; `kbsl` women, both from 2015-16). Game wrappers serve FINISHED games only (an unfinished game raises `wehoop_asset_fetch`). Box scores exist from 2006-07 (no starters flagged that season); play-by-play and shots from 2025-26 only, an earlier finished game raises `wehoop_no_data`. Every game carries a Genius LiveStats id (`genius_id`, joins the `genius` family) from 2018-19 on some seasons and on every game of 2025-26 and later. Birth dates, contact details, photos and logos are dropped. Names and category labels are Turkish; the column descriptions give the English gloss.

Errors: `wehoop_no_data` when TBF has nothing for the request (HTTP 404, an unknown game or player, a season the league list lacks, an empty list, a finished game before 2025-26 asked for play-by-play or shots); `wehoop_bad_request` for an invalid argument or HTTP 400 / 422; `wehoop_asset_fetch` when the fetch failed or the answer is not the one asked for (an unfinished game, a body for another game, season, league or team, an error envelope, a redirect, a partial page or a changed capped list). All inherit `wehoop_error`. Timestamps are Turkey local time (`Europe/Istanbul`).

## Value

A `wehoop_data` tibble. One row per week in play order; `week` is the value the schedule and standings take. Every column below is always present (a zero-row tibble keeps them and their types); an answer with nothing in it raises `wehoop_no_data`, as in sdv-py.

| col_name | type | description |
|---|---|---|
| activity_id | integer | Activity id of the request's league-season (key space: TBF league-seasons). |
| season | integer | Season END year of the request (2027 = 2026-2027). |
| competition | character | TBF league of the request: bsl, tbl or kbsl. |
| week | character | Week number as a string; the value the week argument of tbf_standings and tbf_schedule takes. |
| phase | integer | Phase code (devre = half / phase): 1 = first half of the regular season, 2 = second half, 3 = playoff q |
| phase_name | character | Phase name in Turkish on playoff weeks (Playoff Final = final); null on regular-season weeks. |
| label | character | Week label: 'Hafta N' (week N) or the playoff round name. |
| is_current | logical | True on the current (or last played) week; exactly one row. |

## See Also

Other TBF Functions: `tbf_daily_matches()`, `tbf_game_header()`, `tbf_game_leaders()`, `tbf_game_players()`, `tbf_highlights()`, `tbf_leaders_assists()`, `tbf_leaders_blocks()`, `tbf_leaders_efficiency()`, `tbf_leaders_minutes()`, `tbf_leaders_points()`, `tbf_leaders_rebounds()`, `tbf_leaders_steals()`, `tbf_leaders_threes()`, `tbf_leaders_turnovers()`, `tbf_league_seasons()`, `tbf_leagues()`, `tbf_pbp()`, `tbf_period_scores()`, `tbf_player_box()`, `tbf_player_career()`, `tbf_players()`,

tbf_schedule(), tbf_seasons(), tbf_shots(), tbf_standings(), tbf_team_box(), tbf_team_games(), tbf_team_player_stats(), tbf_team_roster(), tbf_team_staff(), tbf_teams()

## Examples

```
try(tbf_weeks(competition = "bsl", season = 2027))
```

---

update_wbb_db **Build/update wehoop WBB play-by-play database**

---

## Description

update_wbb_db() updates or creates a database with wehoop play by play data of all completed and available games since 2002.

## Usage

```
update_wbb_db(
  dbdir = ".",
  dbname = "wehoop_db",
  tblname = "wehoop_wbb_pbp",
  force_rebuild = FALSE,
  db_connection = NULL,
  datasets = NULL,
  seasons = TRUE
)
```

## Arguments

| | |
|---|---|
| dbdir | Directory in which the database is or shall be located |
| dbname | File name of an existing or desired SQLite database within dbdir |
| tblname | The name of the play by play data table within the database |
| force_rebuild | Hybrid parameter (logical or numeric) to rebuild parts of or the complete play by play data table within the database (please see details for further information) |
| db_connection | A DBIConnection object, as returned by DBI::dbConnect() (please see details for further information) |
| datasets | Optional character vector of dataset shortnames. When NULL (the default) the legacy behavior is preserved: the play-by-play table tblname is created/updated. When supplied, only the requested datasets are written (one table per dataset, named wbb_<dataset> – e.g. wbb_rosters, wbb_player_stats). Valid values: "pbp", "team_box", "player_box", "schedule", "rosters", "player_stats", "team_stats", "standings", "shots", "game_rosters", "officials". |
| seasons | Optional integer vector of seasons. Forwarded to the dataset loaders only when datasets is supplied. Defaults to all available seasons for the dataset (seasons = TRUE). |

### Details

This function creates and updates a data table with the name `tblname` within a SQLite database (other drivers via `db_connection`) located in `dbdir` and named `dbname`. The data table combines all play by play data for every available game back to the 2002 season and adds the most recent completed games as soon as they are available for `wehoop`.

The argument `force_rebuild` is of hybrid type. It can rebuild the play by play data table either for the whole wehoop era (with `force_rebuild = TRUE`) or just for specified seasons (e.g. `force_rebuild = c(2019, 2020)`). Please note the following behavior:

- `force_rebuild = TRUE`: The data table with the name `tblname` will be removed completely and rebuilt from scratch. This is helpful when new columns are added during the Off-Season.
- `force_rebuild = c(2019, 2020)`: The data table with the name `tblname` will be preserved and only rows from the 2019 and 2020 seasons will be deleted and re-added. This is intended to be used for ongoing seasons because ESPN's data provider can make changes to the underlying data during the week.

The parameter `db_connection` is intended for advanced users who want to use other DBI drivers, such as MariaDB, Postgres or odbc. Please note that the arguments `dbdir` and `dbname` are dropped in case a `db_connection` is provided but the argument `tblname` will still be used to write the data table into the database.

### Value

Logical TRUE/FALSE

---

## `update_wnba_db` Build/update wehoop WNBA play-by-play database

---

### Description

update_wnba_db() updates or creates a database with wehoop play by play data of all completed and available games since 2002.

### Usage

```
update_wnba_db(
  dbdir = ".",
  dbname = "wehoop_db",
  tblname = "wehoop_wnba_pbp",
  force_rebuild = FALSE,
  db_connection = NULL,
  datasets = NULL,
  seasons = TRUE
)
```

## Arguments

| | |
|---|---|
| dbdir | Directory in which the database is or shall be located |
| dbname | File name of an existing or desired SQLite database within dbdir |
| tblname | The name of the play by play data table within the database |
| force_rebuild | Hybrid parameter (logical or numeric) to rebuild parts of or the complete play by play data table within the database (please see details for further information) |
| db_connection | A DBIConnection object, as returned by DBI::dbConnect() (please see details for further information) |
| datasets | Optional character vector of dataset shortnames. When NULL (the default) the legacy behavior is preserved: the play-by-play table tblname is created/updated. When supplied, only the requested datasets are written (one table per dataset, named wnba_<dataset> – e.g. wnba_rosters, wnba_player_stats). Valid values: "pbp", "team_box", "player_box", "schedule", "rosters", "player_stats", "team_stats", "standings", "draft", "shots", "game_rosters", "officials". |
| seasons | Optional integer vector of seasons. Forwarded to the dataset loaders only when datasets is supplied. Defaults to all available seasons for the dataset (seasons = TRUE). |

## Details

This function creates and updates a data table with the name tblname within a SQLite database (other drivers via db_connection) located in dbdir and named dbname. The data table combines all play by play data for every available game back to the 2002 season and adds the most recent completed games as soon as they are available for wehoop.

The argument force_rebuild is of hybrid type. It can rebuild the play by play data table either for the whole wehoop era (with force_rebuild = TRUE) or just for specified seasons (e.g. force_rebuild = c(2019, 2020)). Please note the following behavior:

- force_rebuild = TRUE: The data table with the name tblname will be removed completely and rebuilt from scratch. This is helpful when new columns are added during the Off-Season.
- force_rebuild = c(2019, 2020): The data table with the name tblname will be preserved and only rows from the 2019 and 2020 seasons will be deleted and re-added. This is intended to be used for ongoing seasons because ESPN's data provider can make changes to the underlying data during the week.

The parameter db_connection is intended for advanced users who want to use other DBI drivers, such as MariaDB, Postgres or odbc. Please note that the arguments dbdir and dbname are dropped in case a db_connection is provided but the argument tblname will still be used to write the data table into the database.

## Value

Logical TRUE/FALSE

update_wnba_stats_db **Build/update wehoop WNBA Stats database**

## Description

update_wnba_stats_db() writes one or more WNBA Stats API datasets into a database. Mirrors the WNBA / WBB update_*_db() shape but points at the load_wnba_stats_* family. Each dataset is written to its own table, named wnba_stats_<dataset>.

## Usage

```
update_wnba_stats_db(
  dbdir = ".",
  dbname = "wehoop_db",
  datasets = NULL,
  seasons = TRUE,
  db_connection = NULL
)
```

## Arguments

| | |
|---|---|
| dbdir | Directory in which the database is or shall be located. |
| dbname | File name of an existing or desired SQLite database within dbdir. |
| datasets | Character vector of dataset shortnames (see Details). |
| seasons | Optional integer vector of seasons to load. Defaults to all available seasons (seasons = TRUE). |
| db_connection | A DBIConnection object. When supplied, dbdir and dbname are ignored. |

## Details

Unlike update_wnba_db() / update_wbb_db(), there is no historical "play-by-play table" default to preserve, so datasets is required.

Valid datasets values: "schedule", "player_game_logs", "pbp", "rosters", "coaches", "player_stats", "lineups", "team_stats", "standings", "draft", "shots", "game_rosters", "officials", "possessions".

## Value

Invisibly returns TRUE on success.

## wbb_player_crosswalk Get the WBB cross-source player crosswalk

### Description

Build a wide, one-row-per-player-per-team crosswalk linking ESPN and Fox Sports (Bifrost) WBB player identities for a season. ESPN is the anchor source; Fox is matched by normalized name (exact first, then Jaro-Winkler fuzzy with jersey tiebreaker) within each team block. Yahoo columns are NA placeholders. Torvik and the WNBA Stats API have no per-player tables for WBB, so neither source is joined.

### Usage

```
wbb_player_crosswalk(season = most_recent_wbb_season(), min_confidence = 0.92)
```

### Arguments

| | |
|---|---|
| season | Season year (4-digit, e.g. 2025). Defaults to most_recent_wbb_season(). |
| min_confidence | Jaro-Winkler similarity floor for fuzzy matches (default 0.92). |

### Value

A wehoop_data tibble, one row per player per team (ESPN-anchored):

See the HTML help or pkgdown reference for the column table.

### See Also

Other WBB Crosswalk Functions: load_wnba_team_crosswalk(), wbb_schedule_crosswalk(), wbb_team_crosswalk()

### Examples

```
try(wbb_player_crosswalk(season = 2025))
```

## wbb_schedule_crosswalk Get the WBB cross-source schedule crosswalk

### Description

Build a wide, one-row-per-game crosswalk linking ESPN and Bart Torvik (barttorvik.com/ncaaw) game identifiers for a WBB season. Fox Sports and Yahoo game IDs are NA placeholders. Dates are reduced to Eastern-Time game dates before joining; Torvik team1/team2 are unordered (the join uses a sorted team-pair key, so home/away from the Torvik side is not preserved). Games where either Torvik team name cannot be resolved to an ESPN id via wbb_team_crosswalk() are kept as bart_only rows.

### Usage

```
wbb_schedule_crosswalk(season = most_recent_wbb_season())
```

### Arguments

| | |
|---|---|
| season | Season year (4-digit, e.g. 2025). Defaults to most_recent_wbb_season(). |

### Value

A wehoop_data tibble, one row per game:

See the HTML help or pkgdown reference for the column table.

### See Also

Other WBB Crosswalk Functions: load_wnba_team_crosswalk(), wbb_player_crosswalk(), wbb_team_crosswalk()

### Examples

```
try(wbb_schedule_crosswalk(season = 2025))
```

---

## wbb_team_crosswalk Get the WBB cross-source team crosswalk

---

### Description

Build a wide, one-row-per-team-per-season crosswalk linking ESPN, Fox Sports (Bifrost), and Bart Torvik (barttorvik.com/ncaaw) women's college basketball team identities, keyed on espn_team_id. Yahoo columns are NA placeholders. ESPN is deduped by team_id (first occurrence kept). Fox is joined on the full normalized mascot name (with a curated alias bridge for cases where Fox and ESPN differ); Torvik is joined on the normalized school/location name after a curated alias pass for common divergences (e.g. "UConn" / "Connecticut", "Ole Miss" / "Mississippi").

Every source is read **as of** season:

- espn_conference is the conference each team was in that season, under that season's name, from the SDV conference reference (load_wbb_team_group_seasons() and load_wbb_group_seasons()). The ESPN team list itself is today's Division I list, so a team that was not in a Division I conference that season has an NA espn_conference.
- fox_section comes from Fox's per-conference standings for that season (league/standings?groupId=&season=), which start in 2018-19: earlier seasons get NA fox_*. Fox lists teams under the conference they joined the NEXT season, so fox_section is set to NA where it disagrees with espn_conference, and for any Fox conference with fewer than two agreeing teams that stay put the next season. fox_team_id is kept.
- bart_* comes from Torvik's /ncaaw/{season}_team_results.csv (2021 on; earlier seasons get NA bart_*).

A source that fails raises an error instead of returning a crosswalk whose columns are silently all NA. Torvik answering with no teams (blocked or empty) for a season it covers, Fox returning no standings for the season, and a missing conference reference raise an error of class `crosswalk_source_error`.

## Usage

```
wbb_team_crosswalk(season = most_recent_wbb_season(), fox = NULL)
```

## Arguments

| | |
|---|---|
| `season` | Season year (4-digit, ending year, e.g. `2025` = 2024-25). Defaults to `most_recent_wbb_season()`. |
| `fox` | An already-fetched frame with `fox_team_id`, `fox_team_name` and `fox_section`, or `NULL` (default) to fetch `season`'s Fox standings live. Pass an empty `data.frame()` to skip Fox. |

## Value

A `wehoop_data` tibble, one row per ESPN team:

See the HTML help or pkgdown reference for the column table.

## See Also

Other WBB Crosswalk Functions: `load_wnba_team_crosswalk()`, `wbb_player_crosswalk()`, `wbb_schedule_crosswalk()`

## Examples

```
try(wbb_team_crosswalk(season = 2025))
```

---

# wbbl_competition — Women's British Basketball League (Genius LiveStats)

---

## Description

Convenience wrappers binding the Genius family to the Women's British Basketball League (women's; Genius ORG `WBBL`, LiveStats ORG `WBBL`). Season rule: END year; the label span 2025-26 = 2026.

- `wbbl_competitions()`: every competition of the `WBBL` org, with its END-year season (`genius_competitions()`).
- `wbbl_schedule()`: every match of one season (`genius_schedule()`).
- `wbbl_team_box()`, `wbbl_player_box()`, `wbbl_pbp()`, `wbbl_shots()`: one finished game (`genius_team_box()`, `genius_player_box()`, `genius_pbp()`, `genius_shots()`).

## Usage

```
wbbl_competitions(return_parsed = TRUE)

wbbl_schedule(season = NULL, comp_id = NULL, round_number = -1, return_parsed = TRUE)

wbbl_team_box(match_id, return_parsed = TRUE)

wbbl_player_box(match_id, return_parsed = TRUE)

wbbl_pbp(match_id, return_parsed = TRUE)

wbbl_shots(match_id, return_parsed = TRUE)
```

## Arguments

| | |
|---|---|
| season | END-year season (see the season rule). NULL (default) = the newest season the org lists. |
| comp_id | Genius competition id; skips the season lookup (needed when the label rule matches several competitions). |
| round_number | Round to list; -1 (default) lists every round. |
| match_id | Genius match id (match_id of wbbl_schedule()) of a game of this competition. |
| return_parsed | Parse into a tibble (default TRUE); FALSE returns the raw list. |

## Details

The season's competition is picked from the WBBL chooser by the label rule include = Championship, exclude = Play-?Offs (case-insensitive); cups and play-offs have their own comp_id. Several matching competitions raise wehoop_bad_request (pass comp_id); none raises wehoop_no_data.

Genius match ids are one global sequence shared by every customer and data.json names no competition, so each game alias checks the id on the game's LiveStats shell (/u/WBBL/<match_id>/: one extra request per game, cached per process for an hour): its competition name must pass the label rule. Another org's game, a cup or play-off game of WBBL, or a shell request that fails twice (or names no competition) raises wehoop_asset_fetch. The base genius_*() function reads any game unchecked.

Access, proxy and error notes: see genius_game().

```
wbbl_competitions()
wbbl_schedule()
```

## Value

The base function's wehoop_data tibble, same columns and types (the raw list when return_parsed = FALSE), one table per function:

wbbl_competitions():

The columns of genius_competitions():

| col_name | type | description |
|---|---|---|
| org | character | ORG code the request used (the hosted ORG, or the LiveStats ORG for the /u/ shell). |
| comp_id | integer | Genius competition id (compId): one per org competition-season; the {compId} of hosted and data/ |
| label | character | Competition label as the org typed it, free text in any language: may or may not carry a season ('BA |
| league_id | integer | Genius league id of the org (ESAKE = 115; IBBA has two: 187 and 549) |
| year | integer | Genius season year of the competition: the START year of a two-year season (BNXT '2023-2024' = |
| season | integer | sdv END-year season of the competition (2026 = 2025-26) from the label span, else the Genius yea |
| season_rule | character | Which rule produced season: 'label span', 'genius year + 1 (split)', 'genius year (calendar)', 'label |
| selected | logical | True for the competition the hosted chooser pre-selects (the org's default, not always the newest). |

`wbbl_schedule()`:

The columns of `genius_schedule()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | From the block id `extfix_<id>`. Genius match id: the {matchId} of data.json and /u/; one gl |
| link_match_id | integer | Match id of the block's `/match/<id>/summary` link; equals match_id (the capture refuses a p |
| comp_id | integer | CompId of the match link, or of the team links when the org hides match pages (FFBB); equa |
| org | character | ORG code of the match link (or the team links, as comp_id) |
| status | character | Status from the block class STATUS_*: COMPLETE, SCHEDULED, POSTPONED, ... |
| date_time | character | Tip-off as displayed, 'Mon D, YYYY, h:mm AM' (local time, zone not stated); null when un |
| venue_id | integer | Genius venue id (the /venue/{id} of hosted pages); null when the venue is plain text (FFBB) |
| venue_name | character | Venue name as entered (local script) |
| home_team_id | integer | Home team: Genius team id (warehouse key space, global across orgs and seasons; the /team |
| home_team_name | character | Home team full name |
| home_team_code | character | Home team code (local script possible) |
| home_score | integer | Home points; null before tip-off (typed integer only on a route whose 3 sampled rows were a |
| away_team_id | integer | Away team: Genius team id (warehouse key space, global across orgs and seasons; the /team/ |
| away_team_name | character | Away team full name |
| away_team_code | character | Away team code (local script possible) |
| away_score | integer | Away points; null before tip-off (typed integer only on a route whose 3 sampled rows were al |

`wbbl_team_box()`:

The columns of `genius_team_box()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for, |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedul |
| name | character | Team name as the org enters it |
| name_international | character | Team name in Latin script |
| short_name | character | Short team name (shortName in the feed), in the competition's own script when the c |
| code | character | Team code, three letters as entered (may be in the local script, e.g. Greek 'AEK' in C |
| code_international | character | Team code in Latin script |
| coach | character | Head coach display name |
| score | integer | Team points (current or final); a numeric string in the 2014-15 era sample (WBBL 12 |
| full_score | integer | Team points including overtime; equal to score in every sample (a numeric string in t |
| p1_score | integer | Points scored in period 1 (a numeric string in the 2014-15 era) |

| | | |
|---|---|---|
| p2_score | integer | Points scored in period 2 (a numeric string in the 2014-15 era) |
| p3_score | integer | Points scored in period 3 (a numeric string in the 2014-15 era) |
| p4_score | integer | Points scored in period 4 (a numeric string in the 2014-15 era) |
| ot_score | integer | Points the team scored in overtime; null without overtime. Whether several overtime |
| fouls | integer | Team fouls in the current (last) period |
| timeouts | integer | Timeouts counter of the current period (unverified: remaining vs taken); 0 in every sa |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '22 |
| field_goals_made | integer | Team total: field goals made |
| field_goals_attempted | integer | Team total: field goals attempted |
| field_goals_percentage | double | Team total: field-goal percentage, 0-100, rounded to an integer by the feed |
| three_pointers_made | integer | Team total: three-pointers made |
| three_pointers_attempted | integer | Team total: three-pointers attempted |
| three_pointers_percentage | double | Team total: three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Team total: two-pointers made |
| two_pointers_attempted | integer | Team total: two-pointers attempted |
| two_pointers_percentage | double | Team total: two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Team total: free throws made |
| free_throws_attempted | integer | Team total: free throws attempted |
| free_throws_percentage | double | Team total: free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Team total: defensive rebounds |
| rebounds_offensive | integer | Team total: offensive rebounds |
| rebounds_total | integer | Team total: total rebounds |
| assists | integer | Team total: assists |
| turnovers | integer | Team total: turnovers |
| steals | integer | Team total: steals |
| blocks | integer | Team total: blocks made |
| blocks_received | integer | Team total: own shots blocked by an opponent |
| fouls_personal | integer | Team total: personal fouls committed |
| fouls_on | integer | Team total: fouls drawn |
| points | integer | Team total: points |
| points_from_turnovers | integer | Team total: points off turnovers |
| points_second_chance | integer | Team total: second-chance points |
| points_fast_break | integer | Team total: fast-break points |
| points_in_the_paint | integer | Team total: points in the paint |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no l |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no l |
| bench_points | integer | Team total: points scored by non-starters |
| biggest_lead | integer | Team total: biggest lead, points |
| biggest_scoring_run | integer | Team total: biggest unanswered scoring run, points |
| lead_changes | integer | Team total: lead changes in the game (same value on both teams) |
| times_scores_level | integer | Team total: times the score was tied (same value on both teams) |
| time_leading | double | Team total: minutes the team led, decimal minutes (33.58 = 33 min 35 s) |

| | | |
|---|---|---|
| fouls_team | integer | Team total: team (bench) fouls |
| fouls_total | integer | Team total: all fouls committed (personal + technical + team bench) |
| rebounds_team | integer | Team total: team rebounds (credited to no player) |
| rebounds_team_defensive | integer | Team total: team defensive rebounds |
| rebounds_team_offensive | integer | Team total: team offensive rebounds |
| turnovers_team | integer | Team total: team turnovers (credited to no player, e.g. shot clock) |

`wbbl_player_box()`:

The columns of `genius_player_box()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |
| name | character | Display name ('T. Nichols') |
| first_name | character | Given name in the competition's own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition's own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name ('T. Nichols'), in the local script when the org enters one |
| shirt_number | character | Shirt number as a string ('0' and '00' differ) |
| playing_position | character | Playing position code (PG, SG, SF, PF, C, G, F, ...); empty string when not entered |
| starter | integer | 1 when the player started the game, else 0 |
| captain | integer | 1 for the team captain, else 0; absent in some orgs |
| active | integer | 1 when the player is on court now (at game end: on court at the final whistle), else 0 |
| minutes | character | Minutes as the feed prints them, an 'mm:ss' text (can exceed 59 for a team; '199:60' |
| seconds_played | integer | Seconds played parsed from the printed mm:ss text as minutes * 60 + seconds (so '2 |
| field_goals_made | integer | Field goals the player made in the game (two- and three-pointers together; a count). |
| field_goals_attempted | integer | Field goals the player attempted in the game (two- and three-pointers together; a cou |
| field_goals_percentage | double | Field-goal percentage, 0-100, rounded to an integer by the feed |
| three_pointers_made | integer | Three-pointers made |
| three_pointers_attempted | integer | Three-pointers attempted |
| three_pointers_percentage | double | Three-point percentage, 0-100, rounded to an integer |
| two_pointers_made | integer | Two-pointers made |
| two_pointers_attempted | integer | Two-pointers attempted |
| two_pointers_percentage | double | Two-point percentage, 0-100, rounded to an integer |
| free_throws_made | integer | Free throws the player made in the game (a count). |
| free_throws_attempted | integer | Free throws the player attempted in the game (a count). |
| free_throws_percentage | double | Free-throw percentage, 0-100, rounded to an integer |
| rebounds_defensive | integer | Defensive rebounds |
| rebounds_offensive | integer | Offensive rebounds |
| rebounds_total | integer | Total rebounds (offensive plus defensive) the player collected in the game. |
| assists | integer | Assists credited to the player in the game. |
| turnovers | integer | Turnovers committed by the player in the game. |
| steals | integer | Steals credited to the player in the game. |
| blocks | integer | Shots the player blocked in the game (blocks made, not received). |
| blocks_received | integer | Own shots blocked by an opponent |

| | | |
|---|---|---|
| fouls_personal | integer | Personal fouls committed |
| fouls_on | integer | Fouls drawn: personal fouls committed against the player in the game. |
| points | integer | Points the player scored in the game, free throws included. |
| points_from_turnovers | integer | Points off turnovers |
| points_second_chance | integer | Second-chance points |
| points_fast_break | integer | Fast-break points |
| points_in_the_paint | integer | Points the player scored in the paint in the game (the feed's sPointsInThePaint). |
| plus_minus_points | integer | Plus/minus: team point margin while the player was on court |
| eff_1 | integer | Genius efficiency formula 1. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_2 | integer | Genius efficiency formula 2. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_3 | double | Genius efficiency formula 3. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_4 | double | Genius efficiency formula 4. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_5 | integer | Genius efficiency formula 5. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_6 | integer | Genius efficiency formula 6. The feed carries seven formulas (eff_1..eff_7) with no |
| eff_7 | integer | Genius efficiency formula 7. The feed carries seven formulas (eff_1..eff_7) with no |

`wbbl_pbp()`:

The columns of `genius_pbp()`:

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for |
| row_index | integer | Position of the action in the feed order, 0 = the newest action by game time (normall |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted) |
| previous_action | integer | action_number of the linked earlier action (the shot an assist or rebound belongs to); |
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| gt | character | Game clock, 'mm:ss' remaining in the period |
| clock | character | Game clock with hundredths, 'mm:ss:cc' remaining in the period; absent in older fe |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedu |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In- |
| player | character | Player display name; '' when no player |
| shirt_number | character | Player shirt number (string); '' when no player |
| first_name | character | Given name in the competition's own script and spelling (Greek, Hebrew, Czech, ... |
| family_name | character | Family name in the competition's own script |
| international_first_name | character | Given name in Latin script |
| international_family_name | character | Family name in Latin script (often upper case) |
| scoreboard_name | character | Short display name ('T. Nichols'), in the local script when the org enters one |
| action_type | character | Action type: 2pt, 3pt, freethrow, rebound, assist, turnover, steal, block, foul, foulon ( |
| sub_type | character | Action subtype, free-text code per actionType: shot type for 2pt/3pt (layup, drivingla |
| qualifier | character | Qualifier codes of the action joined with commas (pointsinthepaint, fromturnover, 2n |
| success | integer | 1 = made / successful, 0 = missed (shots, free throws); 1 for other actions |
| scoring | integer | 1 when the action is a scoring attempt (shot or free throw), else 0 |
| score_1 | integer | Team 1 score after the action, always an integer (the feed sends a string in most gam |
| score_2 | integer | Team 2 score after the action, always an integer (the feed sends a string in most gam |
| lead | integer | Score margin after the action, team 1 minus team 2 (negative when team 2 leads) |

`wbbl_shots()`:

The columns of genius_shots():

| col_name | type | description |
|---|---|---|
| match_id | integer | Genius match id of the game. data.json does not carry it: it is the match_id asked for, stamped b |
| tno | integer | Team number within this game: 1 = the team listed first (home on the hosted schedule and widg |
| pno | integer | Player number within this game, the key of the team's player map; 0 = no player. In-game only: |
| action_number | integer | Action sequence number, unique within the game (gaps where actions were deleted); a shot join |
| period | integer | Period number; restarts at 1 for the first overtime (read it with period_type). |
| period_type | character | Period type: REGULAR or OVERTIME. |
| action_type | character | Shot value: 2pt or 3pt |
| sub_type | character | Shot type (layup, drivinglayup, jumpshot, pullupjumpshot, dunk, ...) |
| result | integer | Shot result: 1 = made, 0 = missed. |
| x | double | Court x, PERCENT of the full court length (0-100, float), along the long axis. Not normalised to |
| y | double | Court y, PERCENT of the full court width (0-100, float), across the court (observed 1.7-97.2). C |
| player | character | Shooter's display name as the org enters it, in the local script when it uses one. |
| shirt_number | character | Shooter's shirt number (string) |

## See Also

Other Genius Functions: genius_competition_games(), genius_competitions(), genius_game(), genius_game_crosswalk(), genius_match_competition(), genius_pbp(), genius_player_box(), genius_schedule(), genius_schedule_embed(), genius_shots(), genius_team_box(), genius_team_crosswalk(), genius_widget_games(), lfb_competition

---

# wehoop_loaders wehoop Data Loaders Overview

---

## Description

Loaders for full seasons of pre-scraped data from the sportsdataverse/wehoop-{wnba,wbb}-data releases on sportsdataverse-data. Each helper validates the requested seasons, builds the per-asset URLs, downloads in parallel (with optional progressr progress + optional DBI insertion), and tags the result with the wehoop_data S3 class.

## Details

**WNBA loaders:**

| Function | Asset family |
|---|---|
| load_wnba_pbp() | WNBA play-by-play |
| load_wnba_player_box() | WNBA player boxscores |
| load_wnba_team_box() | WNBA team boxscores |
| load_wnba_schedule() | WNBA schedule |
| update_wnba_db() | Delta loader -> DB |

**WBB loaders:**

| Function | Asset family |
|---|---|
| `load_wbb_pbp()` | Women's college basketball PBP |
| `load_wbb_player_box()` | WBB player boxscores |
| `load_wbb_team_box()` | WBB team boxscores |
| `load_wbb_schedule()` | WBB schedule |
| `update_wbb_db()` | Delta loader -> DB |

---

`wjbl_box_score` **Get W League Box Score**

---

## Description

Box score of a final game (2013-14 on): one row per player of both teams.

Endpoint: `GET https://api.wjbl.01core.app/api/boxscore`

## Usage

```
wjbl_box_score(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.

- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_box_score`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across every l |
| team_side | character | Which side of the game the row belongs to: home or away (the API's home / away blocks). |
| team_id | integer | Franchise id team_id of the side (stable across seasons and sponsor renames; 13 = Toyota Ant |
| player_id | integer | Player id player_id (key space: W League persons, STABLE across seasons and teams). The p |
| player_name | character | Player name in Japanese, 'SURNAME GIVEN' with a space (foreign players in katakana). |
| player_number | character | Shirt number, a string. |
| gs | integer | Game started: 1 = starter, 0 = bench (integer). |
| pts | integer | Points. Integer. |
| p2m | integer | Two-point field goals made. Integer. |
| p2a | integer | Two-point field goals attempted. Integer. |
| p3m | integer | Three-point field goals made. Integer. |
| p3a | integer | Three-point field goals attempted. Integer. |
| ftm | integer | Free throws made. Integer. |
| fta | integer | Free throws attempted. Integer. |
| f | integer | Personal fouls committed. Integer. |
| off | integer | Offensive rebounds. Integer. |
| def | integer | Defensive rebounds. Integer. |

| | | |
|---|---|---|
| tot | integer | Total rebounds (off + def). Integer. |
| ast | integer | Assists. Integer. |
| stl | integer | Steals. Integer. |
| blk | integer | Blocked shots (made by this player / team). Integer. |
| to | integer | Turnovers. Integer. |
| min | character | Minutes played 'mm:ss' ('22:19'), a string. |
| min_seconds | integer | Minutes played in whole seconds, from the 'mm:ss' clock string min (null when it does not pa |

## See Also

Other WJBL Functions: `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_box_score(play_schedule_id = 7987))
```

---

`wjbl_game` **Get W League Game**

---

## Description

Game header of a final game: both teams, the result flag and the score by period, one row.

Endpoint: `GET https://api.wjbl.01core.app/api/play-schedule/detail`

## Usage

```
wjbl_game(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.

- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_game`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id: ONE global sequence across every league edition and season (7987 = |
| game_status | character | Game state; always fixed (final) here, the wrapper refuses a game that is not fin |
| home_team_id | integer | Home team: Franchise id team_id (key space: W League franchises, STABLE |
| home_season_team_name | character | Home team name: Team name as registered for the season, in Japanese (compa |
| home_season_team_short_name | character | Home team short name: Short team name for that season, Japanese (Toyota). |
| home_team_logo_image | character | Home team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| home_team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links it |

| | | |
|---|---|---|
| home_season_team_area | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By |
| home_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |
| home_winner_flg | integer | Result for this side: 1 = won, -1 = lost (integer; a 0 tie code is unobserved). |
| home_score | integer | Home points after the play, parsed from score ('HOME-AWAY'); null on plays |
| home_score_p1 | integer | Points scored IN the 1st quarter (not cumulative). |
| home_score_p2 | integer | Points scored in the 2nd quarter. |
| home_score_p3 | integer | Points scored in the 3rd quarter. |
| home_score_p4 | integer | Points scored in the 4th quarter. |
| home_score_ot1 | integer | Points scored in the 1st overtime (5 min); null when there was none. |
| home_score_ot2 | integer | Unobserved: null in every sample. When filled: Points in the 2nd overtime; nu |
| home_score_ot3 | integer | Unobserved: null in every sample. When filled: Points in the 3rd overtime; nul |
| home_score_ot4 | integer | Unobserved: null in every sample. When filled: Points in the 4th overtime; nul |
| away_team_id | integer | Away team: Franchise id team_id (key space: W League franchises, STABLE |
| away_season_team_name | character | Away team name: Team name as registered for the season, in Japanese (compa |
| away_season_team_short_name | character | Away team short name: Short team name for that season, Japanese (Toyota). |
| away_team_logo_image | character | Away team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| away_team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links it |
| away_season_team_area | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By |
| away_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |
| away_winner_flg | integer | Result for this side: 1 = won, -1 = lost (integer; a 0 tie code is unobserved). |
| away_score | integer | Away points after the play, parsed from score ('HOME-AWAY'); null on plays |
| away_score_p1 | integer | Points scored IN the 1st quarter (not cumulative). |
| away_score_p2 | integer | Points scored in the 2nd quarter. |
| away_score_p3 | integer | Points scored in the 3rd quarter. |
| away_score_p4 | integer | Points scored in the 4th quarter. |
| away_score_ot1 | integer | Points scored in the 1st overtime (5 min); null when there was none. |
| away_score_ot2 | integer | Unobserved: null in every sample. When filled: Points in the 2nd overtime; nu |
| away_score_ot3 | integer | Unobserved: null in every sample. When filled: Points in the 3rd overtime; nul |
| away_score_ot4 | integer | Unobserved: null in every sample. When filled: Points in the 4th overtime; nul |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_game(play_schedule_id = 7987))
```

---

`wjbl_game_info` **Get W League Game Info**

---

## Description

Game information of a final game: venue, times, referees, links and the Japanese game report, one row.

Endpoint: `GET https://api.wjbl.01core.app/api/play-schedule`

## Usage

```
wjbl_game_info(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by

[redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.

- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_game_info`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequenc |
| league_id | integer | Competition-edition id league_id (key space: W League competitions; sequent |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25- |
| season_id | integer | Season-phase id season_id (key space: phases WITHIN a league_id, e.g. 169 = |
| season_title | character | Phase title in Japanese: regular season, (Japanese text) / playoff final, playoff s |
| season_full_name | character | league_title + phase (+ division) in Japanese: 2025-26 regular season, Premier |
| game_date | character | Game date in Japanese, (Japanese text) (Saturday 4 April 2026). |
| game_day | character | Game date as an ISO date (YYYY-MM-DD, Japan time), built from the game |
| game_start_time | character | Scheduled tip-off 'HH:MM', Japan time (JST, UTC+9). |
| started_at | character | Actual tip-off 'HH:MM', JST, as recorded ('15:01'); junk on 2013-14 games ( |
| opening_time | character | Doors-open time 'HH:MM', JST; empty when not published. |
| venue | character | Arena name, Japanese. |
| game_icon | integer | Icon code shown next to the game: 0 = none, 1 = home-game icon (home.svg), |
| venue_address | character | Arena postal address, Japanese. |
| venue_others | character | Access directions to the arena (nearest station, walking time), Japanese free tex |
| allow_shoes_on | integer | Outdoor shoes allowed in the arena: 1 = not stated, 2 = yes (o), 3 or 4 = no (x). |
| allow_shoes_on_others | character | Extra note on outdoor shoes; empty string in every sample. |
| ticket_info | character | Ticket information, Japanese free text with CRLF line breaks and HTML links |
| ticket_url | character | Ticket sale URL; empty string in every sample. |
| event_info | character | Event-day information (gate times, food, shoes, tip-off), Japanese free text witl |
| live_link | character | Live-stream URL; empty when none. |
| vod_link | character | Video-on-demand URL (Basketball LIVE / W-TV); empty when none. |
| bs_url | character | Satellite-TV (BS) broadcast URL; empty string in every sample. |
| cs_url | character | Cable/satellite (CS) broadcast URL; empty string in every sample. |
| ground_url | character | Terrestrial-TV broadcast URL; empty string in every sample. |
| other_url | character | Other broadcast URL; empty string in every sample. |
| cancel_flg | integer | Cancelled flag: 1 = cancelled, 0 = not. |
| delay_flg | integer | Postponed flag: 1 = postponed, 0 = not. |

| | | |
|---|---|---|
| game_status | character | Game state: 'fixed' = final (confirmed result), 'scheduled' = not played yet (see |
| box_flg | integer | Box-score flag (integer). 1 on every sampled game, scheduled ones included, s |
| pbp_flg | integer | Play-by-play flag: 1 = /pbp/* routes have data (2018-19 onward), 0 = not (they |
| referee_name | character | Crew chief, 'SURNAME GIVEN' in Japanese; empty on 2013-14 (names are i |
| sub_referee_name1 | character | First umpire, Japanese name; empty when not recorded. |
| sub_referee_name2 | character | Second umpire, Japanese name; empty when not recorded. |
| comment | character | Officials block, Japanese free text: (Japanese text) crew chief, (Japanese text) f |
| report | character | Game report (quarter-by-quarter narrative) in Japanese HTML; filled on 2013- |
| home_team_id | integer | Home team: Franchise id team_id (key space: W League franchises, STABLE |
| home_season_team_name | character | Home team name: Team name as registered for the season, in Japanese (compa |
| home_season_team_short_name | character | Home team short name: Short team name for that season, Japanese (Toyota). |
| home_team_logo_image | character | Home team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| home_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |
| away_team_id | integer | Away team: Franchise id team_id (key space: W League franchises, STABLE |
| away_season_team_name | character | Away team name: Team name as registered for the season, in Japanese (compa |
| away_season_team_short_name | character | Away team short name: Short team name for that season, Japanese (Toyota). |
| away_team_logo_image | character | Away team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| away_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_game_info(play_schedule_id = 7987))
```

---

# wjbl_game_months Get W League Game Months

## Description

Months (YYYY-MM) that have games in a season, newest first; each is one page of wjbl_schedule.

Endpoint: `GET https://api.wjbl.01core.app/api/play-schedule-months`

## Usage

```
wjbl_game_months(season)
```

## Arguments

| | |
|---|---|
| season | Season END year (2026 = 2025-26, 2000 = 1999-2000); the wrapper maps it to the league edition id (league_id) with the live league list (wjbl_leagues, one extra cached request). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_game_months`):

| col_name | type | description |
|---|---|---|
| league_id | integer | Competition edition id league_id, cast to an integer (the API echoes a string here and an integ |
| season | integer | Season END year of the request (2026 = 2025-26); the body itself does not carry it. |
| game_month | character | Month 'YYYY-MM': the game_month parameter of /play-schedules. |
| game_month_jpn | character | The month in Japanese, (Japanese text) (April 2026). |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_game_months(season = 2026))
```

---

# wjbl_head_to_head Get W League Head-to-Head

---

## Description

Head-to-head results grid (hoshitori-hyo) of one or more season phases: one row per game played.

Endpoint: `GET https://api.wjbl.01core.app/api/team-standings`

## Usage

```
wjbl_head_to_head(season_ids)
```

## Arguments

| | |
|---|---|
| season_ids | Season phase id or ids (season_id in wjbl_standing_seasons; a phase inside a league edition, not a year): an integer, a list or a comma string such as '169,170' (2025-26 regular season Premier and Future divisions). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.

- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_head_to_head(season_ids = "169")
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_head_to_head`):

| col_name | type | description |
|---|---|---|
| season_id | integer | Season phase id of the grid, e.g. 169 (2025-26 regular season, Premier division). |
| team_id | integer | Franchise id team_id of the ROW team (the team whose points are win). |
| opponent_team_id | integer | Franchise id team_id of the opponent in that game. |
| game_number | integer | 1-based order of the game within the pair of teams in the grid (the wire list has no number). |
| win | integer | The ROW team's points in that game (a string on the wire, whatever the result; 56 on a loss). |

| | | |
|---|---|---|
| losg | integer | The OPPONENT's points in that game (a string on the wire; the wire key is spelled losg). |
| result | character | 'o 63-56' = row team won, 'x 56-63' = lost (row team's points first); 'o 20-0' is a forfeit win |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

---

`wjbl_leagues` **Get W League Leagues**

---

## Description

Every competition edition (league_id): the W League seasons back to 1999-2000, all-star games, cups and camps.

Endpoint: `GET https://api.wjbl.01core.app/api/leagues`

## Usage

```
wjbl_leagues()
```

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.

- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_leagues`):

| col_name | type | description |
|---|---|---|
| league_id | integer | Competition-edition id league_id (key space: W League competitions; sequential, ONE id per seaso |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League (sdv END year 2026 |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25-26 W League'); null |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_leagues())
```

`wjbl_pbp` **Get W League Play-by-Play**

## Description

Play-by-play of a final game (2018-19 on): one row per play, Japanese action vocabulary, no shot x/y.

Endpoint: `GET https://api.wjbl.01core.app/api/pbp/play-by-play`

## Usage

```
wjbl_pbp(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19

(earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.

- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_pbp`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across every |
| no | integer | Play sequence number, ascending, 1 = first; numbers can skip (deleted plays: 443 numbers f |
| period | integer | Period: 1-4 = quarters, 5-14 = overtime 1-10 (the site's tab map). Integer. |
| rest_time | character | Game clock REMAINING in the period, 'm:ss' without a leading zero ('9:38'; '10:00' at th |
| seconds_remaining | integer | Seconds remaining in the period, from the 'm:ss' clock string rest_time. |
| team_id | integer | Acting team: franchise id team_id; null on period and game-end lines. |
| team_name | character | Acting team name, Japanese (may contain a full-width space and differ from the box score's |
| team_side | character | Side of the acting team, home or away (from home_away: 1 home, 2 away); null on period a |
| player_number1 | character | Shirt number of the first actor (the player going OUT on a substitution), a string; '' on team a |
| player_number2 | character | Shirt number of the second actor (the player coming IN on a substitution), a string; '' otherw |
| player_name1 | character | Name of the first actor, Japanese; '' on team and period lines. |
| player_name2 | character | Name of the second actor (substitution in), Japanese; '' otherwise. |
| player_id1 | integer | First actor (the player going OUT on a substitution): Player id (key space: W League person |
| player_id2 | integer | Second actor, filled on substitutions only (the player coming IN; verified 89/89 on game 798 |
| play_text | character | Display text in Japanese: (Japanese text). Bracketed running counts: shots the player's point |
| score | character | Score 'HOME-AWAY' after the play ('4-3'); filled on scoring plays and period/game-end li |
| home_score | integer | Home points after the play, parsed from score ('HOME-AWAY'); null on plays that carry no |
| away_score | integer | Away points after the play, parsed from score ('HOME-AWAY'); null on plays that carry no |
| end_period_flg | integer | End-of-period flag (integer); 0 on every observed row, the period end is the (Japanese text) a |
| end_game_flg | integer | End-of-game flag: 1 on the final (Japanese text) line only (the site skips that row), else 0. |
| action1 | character | Action in Japanese (vocabulary of 8 full games, 2018-19 to 2025-26): player in (the starting |
| action2 | character | Sub-action in Japanese: jump shot (on field-goal attempts), free throws awarded on this foul |
| action3 | character | Second qualifier in Japanese: shooting foul; '' otherwise. |
| home_away | integer | Side of the acting team: 1 = home, 2 = away; null on period/game-end lines. |

## See Also

Other WJBL Functions: wjbl_box_score(), wjbl_game(), wjbl_game_info(), wjbl_game_months(), wjbl_head_to_head(), wjbl_leagues(), wjbl_period_scores(), wjbl_player(), wjbl_player_period_leaders(), wjbl_roster(), wjbl_schedule(), wjbl_staff(), wjbl_standing_seasons(), wjbl_team_box_score(), wjbl_team_period_stats(), wjbl_team_ranking(), wjbl_team_standings(), wjbl_teams()

## Examples

```
try(wjbl_pbp(play_schedule_id = 7987))
```

---

# wjbl_period_scores Get W League Period Scores

---

## Description

Points of each side in each period played (quarters and overtimes) from the game header.

Endpoint: `GET https://api.wjbl.01core.app/api/play-schedule/detail`

## Usage

```
wjbl_period_scores(play_schedule_id)
```

## Arguments

| | |
|---|---|
| play_schedule_id | Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final. |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.

- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_period_scores(play_schedule_id = 7987)
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_period_scores`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across every l |
| team_side | character | Which side of the game the row belongs to: home or away (the API's home / away blocks). |
| team_id | integer | Franchise id team_id of the side (stable across seasons and sponsor renames; 13 = Toyota Ant |
| period | integer | Period number: 1-4 are the quarters and 5-8 the first to fourth overtime (score_ot1 to score_ot |
| period_label | character | Readable period: Q1-Q4 quarters, OT1-OT10 overtimes (period categories 5-14), H1 / H2 hal |
| score | integer | Points the side scored in that period (null periods, such as overtimes not played, have no row). |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

`wjbl_player` **Get W League Player**

## Description

Player profile: current team, names, bio and career notes, one row.

Endpoint: `GET https://api.wjbl.01core.app/api/player`

## Usage

```
wjbl_player(player_id)
```

## Arguments

| | |
|---|---|
| `player_id` | Player id (player_id in wjbl_box_score or wjbl_roster): the person, stable across seasons, e.g. 9799. |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.

- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_player(player_id = 9799)
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_player`):

| col_name | type | description |
|---|---|---|
| player_id | integer | Player id player_id (key space: W League persons, STABLE across seasons and te |
| player_name | character | Name in Japanese, 'SURNAME GIVEN'. |
| player_name_en | character | Name in romaji, uppercase, 'SURNAME GIVEN' ('OKAMOTO MIYUU'). |
| player_furigana | character | Name reading in hiragana ((Japanese text)). |
| team_id | integer | Current (or last) team: Franchise id team_id (key space: W League franchises, ST |
| season_team_name | character | Team name as registered for the season, in Japanese (company + nickname: Toyot |
| season_team_short_name | character | Short team name for that season, Japanese (Toyota). |
| season_team_player_number | character | Shirt number this season, a string. |
| team_logo_image | character | Team logo path, relative to https://www.wjbl.org ('/static/images/base_team/...png |
| team_url | character | Team's own website URL. |
| player_birthday | character | Date of birth 'YYYY/MM/DD' (public player bio). |
| player_nickname | character | Nickname, Japanese. |
| player_position | character | Playing position code ('PG', 'SG', 'SF', 'PF', 'C'). |
| player_home_town | character | Home prefecture, Japanese (Hokkaido). |
| player_height | character | Height with unit, a string ('179cm'). |
| player_weight | character | Weight with unit, a string ('77kg'). |
| player_height_cm | integer | Height in whole centimetres, parsed from player_height ('179cm'). |
| player_weight_kg | integer | Weight in whole kilograms, parsed from player_weight ('77kg'). |
| player_affiliation_history | character | Career path, Japanese, (Japanese text)-separated (mini-basketball club (Japanese) |
| player_independent_flg | integer | Retired flag (the site greys the team and hides the number): 1 = retired / no longer |
| player_national_team | character | National-team history, Japanese free text (Universiade). |
| player_awards | character | Awards and honours, Japanese free text. |
| player_coach_history | character | Coaching history (for players who became coaches); empty string in the sample. |
| player_coach_record | character | Coaching record; empty string in the sample. |
| player_picture | character | Photo path relative to https://www.wjbl.org. |

| | | |
|---|---|---|
| player_wjbl_flg | integer | 1 = a W League player (0 presumably for others; unverified). |
| player_rookie | integer | Rookie ((Japanese) icon): 1 = yes, 0 = no. |
| player_early_entry | integer | Early-entry player ((Japanese), a student who joined before graduating): 1 = yes, 0 |
| player_foreign_national_flg | integer | Foreign national ((Japanese) icon): 1 = yes, 0 = no. |
| player_naturalized_flg | integer | Naturalised player ((Japanese) icon): 1 = yes, 0 = no. |
| player_appeal | character | Self-introduction text; empty string in the sample. |
| player_championships | character | Championships text; empty string in the sample. |
| player_url | character | Personal website URL; empty string in the sample. |
| player_facebook_url | character | Facebook URL; empty string in the sample. |
| player_twitter_url | character | X (Twitter) URL; empty string in the sample. |
| player_instagram_url | character | Instagram URL; empty string in the sample. |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

---

`wjbl_player_period_leaders`

# Get W League Player Period Leaders

---

## Description

Leading scorer, passer and rebounder of each team per period, half and game (2018-19 on).

Endpoint: `GET https://api.wjbl.01core.app/api/pbp/player-stats`

## Usage

```
wjbl_player_period_leaders(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt

(404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.

- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_player_period_leaders(play_schedule_id = 7987)
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_player_period_leaders`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across ev |
| home_team_id | integer | Home team: Franchise id team_id (key space: W League franchises, STABLE across sea |
| away_team_id | integer | Away team: Franchise id team_id (key space: W League franchises, STABLE across sea |

| | | |
|---|---|---|
| period_category | integer | The period category again, as an integer (camelCase here, period_category on /pbp/team- |
| period_label | character | Readable period: Q1-Q4 quarters, OT1-OT10 overtimes (period categories 5-14), H1 / H |
| home_point_player | character | Home points leader(s) in the period as '#NUMBER NAME' (Japanese); ties are joined w |
| home_point_value | character | Their points with the counter (Japanese) ((Japanese text)); ties joined with ', ' in the sam |
| home_assist_player | character | Home assist leader(s), '#NUMBER NAME', ties joined with ', '. |
| home_assist_value | character | Their assists with the counter (Japanese) ((Japanese text)), ', '-joined for ties. |
| home_rebound_player | character | Home rebound leader(s), '#NUMBER NAME', ties joined with ', '. |
| home_rebound_value | character | Their rebounds 'offensive-defensive-total' ('0-3-3'), ', '-joined for ties. |
| away_point_player | character | Away points leader(s) in the period as '#NUMBER NAME' (Japanese); ties are joined w |
| away_point_value | character | Their points with the counter (Japanese) ((Japanese text)); ties joined with ', ' in the sam |
| away_assist_player | character | Away assist leader(s), '#NUMBER NAME', ties joined with ', '. |
| away_assist_value | character | Their assists with the counter (Japanese) ((Japanese text)), ', '-joined for ties. |
| away_rebound_player | character | Away rebound leader(s), '#NUMBER NAME', ties joined with ', '. |
| away_rebound_value | character | Their rebounds 'offensive-defensive-total' ('0-3-3'), ', '-joined for ties. |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

---

`wjbl_roster` **Get W League Roster**

---

## Description

Players of a team in a season phase, with the head coach.

Endpoint: `GET https://api.wjbl.01core.app/api/team/roster`

## Usage

```
wjbl_roster(team_id, season_id = NULL)
```

## Arguments

| | |
|---|---|
| `team_id` | Franchise id (team_id in wjbl_teams): stable across seasons and sponsor renames, e.g. 13 (Toyota Antelopes). |
| `season_id` | Season phase id (season_id in wjbl_standing_seasons), e.g. 169 (2025-26 regular season, Premier division). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_roster(season_id = 169, team_id = 13)
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_roster`):

| col_name | type | description |
|---|---|---|
| league_id | integer | Competition-edition id league_id (key space: W League competitions; sequential, ONE id |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League (sdv END y |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25-26 W Leagu |
| season_id | integer | The season_id asked for: Season-phase id season_id (key space: phases WITHIN a league_ |
| season_full_name | character | league_title + phase (+ division) in Japanese: 2025-26 regular season, Premier division (Fu |
| season_team_id | integer | Team-season registration id (key space: one per team per season phase; 1343 = Toyota in s |
| team_id | integer | The team_id asked for: Franchise id team_id (key space: W League franchises, STABLE a |
| coach_id | integer | Head-coach id (key space: W League coaches; the coach_id of /coach and of the site's /coa |
| coach_name | character | Head coach name, Japanese. |
| team_comment | character | Free text the league shows with the roster (captains and other notes, HTML with <br>), as |
| player_id | integer | Player id player_id (key space: W League persons, STABLE across seasons and teams). Th |
| player_name | character | Player name, Japanese. |
| player_number | character | Shirt number, a string. |
| captain_flg | integer | Captain: 1 = yes, 0 = no. |
| sub_captain_flg | integer | Vice-captain: 1 = yes, 0 = no. |
| transfer_flg | integer | Transferred in this season ((Japanese) icon): 1 = yes, 0 = no. |
| position | character | Playing position (G / F / C style code); empty in every sampled row. |
| rookie | integer | Rookie ((Japanese) icon): 1 = yes, 0 = no. |
| early_entry | integer | Early-entry player ((Japanese), a student who joined before graduating): 1 = yes, 0 = no. |
| picture | character | Player photo path relative to https://www.wjbl.org. |
| foreign_national_flg | integer | Foreign national ((Japanese) icon): 1 = yes, 0 = no. |
| naturalized_flg | integer | Naturalised player ((Japanese) icon): 1 = yes, 0 = no. |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

---

`wjbl_schedule` **Get W League Schedule**

---

## Description

Games of a season in ONE month (default: the season's latest month) with teams, venue, status and score.

Endpoint: `GET https://api.wjbl.01core.app/api/play-schedules`

## Usage

```
wjbl_schedule(season, game_month = NULL, team_id = NULL)
```

## Arguments

| | |
|---|---|
| season | Season END year (2026 = 2025-26, 2000 = 1999-2000); the wrapper maps it to the league edition id (league_id) with the live league list (wjbl_leagues, one extra cached request). |
| game_month | Month to list, 'YYYY-MM' (see wjbl_game_months). Omitted = the season's latest month only, NOT the whole season: the API pages by month. |
| team_id | Franchise id (team_id in wjbl_teams): stable across seasons and sponsor renames, e.g. 13 (Toyota Antelopes). Narrows the month to that team's games. |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.

- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_schedule`):

| col_name | type | description |
|---|---|---|
| league_id | integer | The league_id asked for (integer): Competition-edition id league_id (key space |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25- |
| season_id | integer | Season-phase id season_id (key space: phases WITHIN a league_id, e.g. 169 = |
| season_title | character | Phase title in Japanese: regular season, (Japanese text) / playoff final, playoff s |
| season_full_name | character | league_title + phase (+ division) in Japanese: 2025-26 regular season, Premier |
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequenc |
| game_year | integer | Calendar year of the game, 'YYYY' string (not the season); parsed to an intege |
| game_date | character | Game day 'MM.DD' ('04.04'); with game_year it is the date, Japan time. |
| game_day | character | Game day 'MM.DD' ('04.04'); with game_year it is the date, Japan time. |
| game_date_week_day_name | character | Weekday in Japanese, one character: (Japanese) Mon, (Japanese) Tue, (Japanes |
| game_start_time | character | Scheduled tip-off 'HH:MM', Japan time (JST, UTC+9). |
| started_at | character | Actual tip-off 'HH:MM', JST, as recorded ('15:01'); junk on 2013-14 games (' |
| opening_time | character | Doors-open time 'HH:MM', JST; empty when not published. |
| venue | character | Arena name, Japanese. |
| venue_address | character | Arena postal address, Japanese. |
| game_icon | integer | Icon code shown next to the game: 0 = none, 1 = home-game icon (home.svg), |
| ticket_url | character | Ticket sale URL; empty string in every sample. |
| vod_link | character | Video-on-demand URL (Basketball LIVE / W-TV); empty when none. |
| live_link | character | Live-stream URL; empty when none. |
| game_status | character | Game state: 'fixed' = final (confirmed result), 'scheduled' = not played yet (see |
| box_flg | integer | Box-score flag (integer). 1 on every sampled game, scheduled ones included, s |
| pbp_flg | integer | Play-by-play flag: 1 = /pbp/* routes have data (2018-19 onward), 0 = not (they |
| cancel_flg | integer | Cancelled flag: 1 = cancelled, 0 = not. |
| delay_flg | integer | Postponed flag: 1 = postponed, 0 = not. |
| ot_flg | integer | Overtime flag: 1 = the game went to overtime, 0 = not. |
| home_team_id | integer | Home team: Franchise id team_id (key space: W League franchises, STABLE |
| home_season_team_name | character | Home team name: Team name as registered for the season, in Japanese (compa |
| home_season_team_short_name | character | Home team short name: Short team name for that season, Japanese (Toyota). |
| home_team_logo_image | character | Home team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| home_team_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |
| home_team_score | integer | Home final points (integer). Every sampled game is finished, so null is not in th |

| | | |
|---|---|---|
| home_team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links it |
| away_team_id | integer | Away team: Franchise id team_id (key space: W League franchises, STABLE |
| away_season_team_name | character | Away team name: Team name as registered for the season, in Japanese (compa |
| away_season_team_short_name | character | Away team short name: Short team name for that season, Japanese (Toyota). |
| away_team_logo_image | character | Away team logo: Team logo path, relative to https://www.wjbl.org ('/static/ima |
| away_team_free_text | character | Free-text team label used when the slot has no team (e.g. 'winner of game X'); |
| away_team_score | integer | Away final points (integer). Null on unplayed games, as for home_team_score |
| away_team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links it |

## See Also

Other WJBL Functions: wjbl_box_score(), wjbl_game(), wjbl_game_info(), wjbl_game_months(), wjbl_head_to_head(), wjbl_leagues(), wjbl_pbp(), wjbl_period_scores(), wjbl_player(), wjbl_player_period_leaders(), wjbl_roster(), wjbl_staff(), wjbl_standing_seasons(), wjbl_team_box_score(), wjbl_team_period_stats(), wjbl_team_ranking(), wjbl_team_standings(), wjbl_teams()

## Examples

```
try(wjbl_schedule(game_month = "2026-04", season = 2026))
```

---

`wjbl_staff` **Get W League Staff**

---

## Description

Staff of a team in a season phase (general manager, assistant coaches, support staff).

Endpoint: `GET https://api.wjbl.01core.app/api/team/roster`

## Usage

```
wjbl_staff(team_id, season_id = NULL)
```

## Arguments

| | |
|---|---|
| `team_id` | Franchise id (team_id in wjbl_teams): stable across seasons and sponsor renames, e.g. 13 (Toyota Antelopes). |
| `season_id` | Season phase id (season_id in wjbl_standing_seasons), e.g. 169 (2025-26 regular season, Premier division). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_staff(season_id = 169, team_id = 13)
```

## Value

A wehoop_data tibble, one row per record; the columns are always these, in this order (sdv-py wjbl_staff):

| col_name | type | description |
|---|---|---|
| league_id | integer | Competition-edition id league_id (key space: W League competitions; sequential, ONE id pe |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League (sdv END yea |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25-26 W League' |
| season_id | integer | The season_id asked for: Season-phase id season_id (key space: phases WITHIN a league_id |
| season_full_name | character | league_title + phase (+ division) in Japanese: 2025-26 regular season, Premier division (Futu |
| season_team_id | integer | Team-season registration id (key space: one per team per season phase; 1343 = Toyota in seas |
| team_id | integer | The team_id asked for: Franchise id team_id (key space: W League franchises, STABLE acr |
| staff_name | character | Staff member name, Japanese. |
| position | character | Staff role, Japanese: (Japanese) associate head coach, (Japanese) assistant coach, (Japanese) g |
| url | character | Link attached to the staff entry; empty in most rows. |

## See Also

Other WJBL Functions: wjbl_box_score(), wjbl_game(), wjbl_game_info(), wjbl_game_months(), wjbl_head_to_head(), wjbl_leagues(), wjbl_pbp(), wjbl_period_scores(), wjbl_player(), wjbl_player_period_leaders(), wjbl_roster(), wjbl_schedule(), wjbl_standing_seasons(), wjbl_team_box_score(), wjbl_team_period_stats(), wjbl_team_ranking(), wjbl_team_standings(), wjbl_teams()

---

# wjbl_standing_seasons Get W League Standing Seasons

---

## Description

Every season phase (season_id) with its edition name and standings flags.

Endpoint: GET https://api.wjbl.01core.app/api/seasons/standings

## Usage

```
wjbl_standing_seasons()
```

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.

- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_standing_seasons`):

| col_name | type | description |
|---|---|---|
| season_id | integer | Season-phase id season_id (key space: phases WITHIN a league_id, e.g. 169 = 25- |
| season_full_name | character | Edition name with the sponsor, Japanese, e.g. (Japanese text) (truncated by the API |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25-26 W |
| season_league_name | character | Phase / division label in Japanese: Premier, Future, group stage, second round, low |
| team_standings_flg | integer | 1 = the phase has a standings grid (/team-standings), 0 = not (cups, all-star games, p |
| season_tournament_text_flg | integer | 1 = the phase shows a tournament text instead of a table, 0 = not; 0 in every sample |
| season_tournament_text | character | Tournament text (HTML) for knockout phases; empty in every sample. |

## See Also

Other WJBL Functions: wjbl_box_score(), wjbl_game(), wjbl_game_info(), wjbl_game_months(), wjbl_head_to_head(), wjbl_leagues(), wjbl_pbp(), wjbl_period_scores(), wjbl_player(), wjbl_player_period_leaders(), wjbl_roster(), wjbl_schedule(), wjbl_staff(), wjbl_team_box_score(), wjbl_team_period_stats(), wjbl_team_ranking(), wjbl_team_standings(), wjbl_teams()

## Examples

```
try(wjbl_standing_seasons())
```

---

# wjbl_team_box_score Get W League Team Box Score

---

## Description

Team line and totals of each side of a final game's box score, with the shooting percentages.

Endpoint: `GET https://api.wjbl.01core.app/api/boxscore`

## Usage

```
wjbl_team_box_score(play_schedule_id)
```

## Arguments

| | |
|---|---|
| play_schedule_id | Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final. |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.

- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_team_box_score`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across every l |
| team_side | character | Which side of the game the row belongs to: home or away (the API's home / away blocks). |
| team_id | integer | Franchise id team_id of the side (stable across seasons and sponsor renames; 13 = Toyota Ant |
| row_type | character | Which line of the side: team (stats credited to the team itself) or total (the whole side). |
| pts | integer | Points scored by the side (the sum of its players' points). |
| p2m | integer | Two-point field goals made. |
| p2a | integer | Two-point field goals attempted. |
| p3m | integer | Three-point field goals made. |
| p3a | integer | Three-point field goals attempted. |
| ftm | integer | Free throws made. |

| | | |
|---|---|---|
| fta | integer | Free throws attempted. |
| f | integer | Personal fouls committed. |
| off | integer | Offensive rebounds. |
| def | integer | Defensive rebounds. |
| tot | integer | Total rebounds (offensive plus defensive). |
| ast | integer | Assists credited to the side. |
| stl | integer | Steals credited to the side. |
| blk | integer | Blocked shots credited to the side. |
| to | integer | Turnovers committed by the side. |
| min | character | Minutes: summed minutes 'mmm:ss' on the total row ('200:00' = 5 x 40:00), always '00:00' |
| min_seconds | integer | Minutes played in whole seconds, from the 'mm:ss' clock string min (null when it does not pa |
| p2p | double | Two-point field-goal percentage 0-100 (a string on the wire); total row only. |
| p3p | double | Three-point field-goal percentage 0-100 (a string on the wire); total row only. |
| ftp | double | Free-throw percentage 0-100 (a string on the wire); total row only. |
| fgp | double | Field-goal percentage 0-100, two- and three-pointers together (a string on the wire); total row |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

## Examples

```
try(wjbl_team_box_score(play_schedule_id = 7987))
```

---

`wjbl_team_period_stats`

# Get W League Team Period Stats

---

## Description

Team stat summary per period, half and game (2018-19 on): percentages, rebounds, assists, turnovers, lead changes.

Endpoint: `GET https://api.wjbl.01core.app/api/pbp/team-stats`

## Usage

```
wjbl_team_period_stats(play_schedule_id)
```

## Arguments

`play_schedule_id`
: Game id (play_schedule_id in wjbl_schedule): ONE global sequence across every league edition and season, e.g. 7987 (2025-26 playoff final game 1), 5533 (2018-19), 4243 (2013-14 final, box score only). The game must be final.

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_team_period_stats(play_schedule_id = 7987)
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_team_period_stats`):

| col_name | type | description |
|---|---|---|
| play_schedule_id | integer | Game id play_schedule_id (key space: W League games, ONE global sequence across ev |
| home_team_id | integer | Home team: Franchise id team_id (key space: W League franchises, STABLE across sea |
| away_team_id | integer | Away team: Franchise id team_id (key space: W League franchises, STABLE across sea |
| period_category | integer | The period category again, as an integer (see periods). |
| period_label | character | Readable period: Q1-Q4 quarters, OT1-OT10 overtimes (period categories 5-14), H1 / H |
| home_ptr | double | Home field-goal percentage (2PT + 3PT) in the period, 0-100, a NUMBER with up to on |
| home_pt2r | double | Home two-point percentage, 0-100, number. |
| home_pt3r | double | Home three-point percentage, 0-100, number. |
| home_ftr | double | Home free-throw percentage, 0-100, number (0 when no attempts). |
| home_rebound_off | integer | Home offensive rebounds in the period, the first number of the wire's 'off-def-total' strin |
| home_rebound_def | integer | Home defensive rebounds in the period, the second number of the wire's 'off-def-total' s |
| home_rebound_tot | integer | Home total rebounds in the period, the third number of the wire's 'off-def-total' string. |
| home_assist | integer | Home assists (integer). |
| home_turnovers | integer | Home turnovers in the period, from the wire's counted string (6); the wire key is spelled |
| home_max_difference | integer | Home team's largest lead in the period, in points; the wire key is spelled home_max_diff |
| away_ptr | double | Away field-goal percentage (2PT + 3PT) in the period, 0-100, a NUMBER with up to on |
| away_pt2r | double | Away two-point percentage, 0-100, number. |
| away_pt3r | double | Away three-point percentage, 0-100, number. |
| away_ftr | double | Away free-throw percentage, 0-100, number (0 when no attempts). |
| away_rebound_off | integer | Away offensive rebounds in the period, the first number of the wire's 'off-def-total' string |
| away_rebound_def | integer | Away defensive rebounds in the period, the second number of the wire's 'off-def-total' st |
| away_rebound_tot | integer | Away total rebounds in the period, the third number of the wire's 'off-def-total' string. |
| away_assist | integer | Away assists (integer). |
| away_turnovers | integer | Away turnovers in the period, from the wire's counted string (6); the wire key is spelled a |
| away_max_difference | integer | Away team's largest lead in the period, points (integer). |
| lead_count | integer | Lead changes in the period, from the wire's counted string (0 times). |
| tie_count | integer | Times the score was tied in the period, from the wire's counted string (4 times). |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_ranking()`, `wjbl_team_standings()`, `wjbl_teams()`

---

`wjbl_team_ranking` **Get W League Team Ranking**

---

## Description

League table with per-game team averages (points, shooting, rebounds, assists) of one or more season phases.

Endpoint: `GET https://api.wjbl.01core.app/api/team-ranking`

## Usage

```
wjbl_team_ranking(season_ids)
```

## Arguments

| | |
|---|---|
| `season_ids` | Season phase id or ids (season_id in wjbl_standing_seasons; a phase inside a league edition, not a year): an integer, a list or a comma string such as '169,170' (2025-26 regular season Premier and Future divisions). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.

- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_team_ranking(season_ids = "169,170")
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_team_ranking`):

| col_name | type | description |
|---|---|---|
| league_id | integer | Competition-edition id league_id (key space: W League competitions; sequential, ON |
| league_title | character | Competition-edition title in Japanese: '25-26 W League' = 2025-26 W League (sdv E |
| season | integer | Season END year (2025-26 = 2026, 1999-2000 = 2000) from league_title ('25-26 W |
| season_id | integer | Season-phase id season_id (key space: phases WITHIN a league_id, e.g. 169 = 25-26 |
| season_title | character | Phase title in Japanese: regular season, (Japanese text) / playoff final, playoff semifina |
| season_full_name | character | league_title + phase (+ division) in Japanese: 2025-26 regular season, Premier divisio |
| season_league_name | character | Phase / division label in Japanese: Premier, Future, group stage, second round, lower |
| season_league_flg | integer | Division layout flag: 1 = one table per division (Premier / Future), 2 = split by season |
| season_league_img | character | Division badge image path relative to https://www.wjbl.org. |
| season_tournament_text | character | Tournament text (HTML); empty in every sample. |
| update_info | character | As-of note in Japanese, (Japanese text) (as of 8 Feb 2026). |
| team_id | integer | Franchise id team_id (key space: W League franchises, STABLE across seasons and s |
| season_team_name | character | Team name as registered for the season, in Japanese (company + nickname: Toyota A |
| season_team_short_name | character | Short team name for that season, Japanese (Toyota). |
| team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links its roster |
| position | integer | Rank of the team in the phase (a string on the wire). |
| season_team_area | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By the site |
| win | integer | Wins, integer. |
| lost | integer | Losses, integer. |
| winning_rate | double | Winning percentage as a FRACTION 0-1 with three decimals, a string ('0.821'); pars |
| ppg | double | Points per game, a string with two decimals; parsed to a number. |
| opp | double | Opponent points per game (OPP), a string with two decimals; parsed to a number. |
| fg | double | Field-goal percentage 0-100 with a '%' sign, a string ('41.99%'); parsed to a number. |
| opp_fg | double | Opponents' field-goal percentage 0-100 with '%', a string; parsed to a number. |
| p3pg | double | Three-pointers made per game (3PPG), a string; parsed to a number. |
| rpg | double | Rebounds per game, a string; parsed to a number. |
| apg | double | Assists per game, a string; parsed to a number. |
| spg | double | Steals per game, a string; parsed to a number. |
| bpg | double | Blocks per game, a string; parsed to a number. |
| tpg | double | Turnovers per game, a string; parsed to a number. |

| | | |
|---|---|---|
| gpd | double | Point differential summed over the season (GPD), NOT per game: (ppg - opp) x game |
| win_point | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By the site |
| unsatisfied | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By the site |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_standings()`, `wjbl_teams()`

---

# wjbl_team_standings Get W League Team Standings

---

## Description

League table of one or more season phases: rank and record of every team.

Endpoint: `GET https://api.wjbl.01core.app/api/team-standings`

## Usage

```
wjbl_team_standings(season_ids)
```

## Arguments

| | |
|---|---|
| season_ids | Season phase id or ids (season_id in wjbl_standing_seasons; a phase inside a league edition, not a year): an integer, a list or a comma string such as '169,170' (2025-26 regular season Premier and Future divisions). |

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.

- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.
- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_team_standings`):

| col_name | type | description |
|---|---|---|
| season_id | integer | Season-phase id season_id (key space: phases WITHIN a league_id, e.g. 169 = 25-26 |
| season_title | character | Phase title in Japanese: regular season, (Japanese text) / playoff final, playoff semifina |
| league_name | character | Division name, Japanese (Premier). |
| season_league_name | character | Phase / division label in Japanese: Premier, Future, group stage, second round, lower |
| season_league_flg | integer | Division layout flag: 1 = one table per division (Premier / Future), 2 = split by season |
| update_date | character | As-of date in Japanese, (Japanese text). |
| team_id | integer | Franchise id team_id (key space: W League franchises, STABLE across seasons and s |
| season_team_name | character | Team name as registered for the season, in Japanese (company + nickname: Toyota A |
| season_team_short_name | character | Short team name for that season, Japanese (Toyota). |
| team_entry_flg | integer | Official-league participant flag: 1 = a W League member team (the site links its roster |
| position | integer | Rank of the team in the phase (a string on the wire, equal to the map key). |
| season_team_area | character | Unobserved: null in every sample, so no type and no meaning is claimed. (By the site |
| win_lost | character | Record in Japanese, (Japanese text) (23 wins, 5 losses). |
| wins | integer | Wins in the record, parsed from win_lost (23 wins). |
| losses | integer | Losses in the record, parsed from win_lost (5 losses). |

## See Also

Other WJBL Functions: wjbl_box_score(), wjbl_game(), wjbl_game_info(), wjbl_game_months(), wjbl_head_to_head(), wjbl_leagues(), wjbl_pbp(), wjbl_period_scores(), wjbl_player(), wjbl_player_period_leaders(), wjbl_roster(), wjbl_schedule(), wjbl_staff(), wjbl_standing_seasons(), wjbl_team_box_score(), wjbl_team_period_stats(), wjbl_team_ranking(), wjbl_teams()

## Examples

```
try(wjbl_team_standings(season_ids = "169"))
```

---

`wjbl_teams` **Get W League Teams**

---

## Description

Every franchise (team_id) the site knows, all-star sides included.

Endpoint: `GET https://api.wjbl.01core.app/api/teams/base`

## Usage

```
wjbl_teams()
```

## Details

- Unofficial and unsupported: these wrappers read the JSON backend of www.wjbl.org (api.wjbl.01core.app), keyless, found in the site's own JavaScript. The site's terms of use (www.wjbl.org/kiyaku/) forbid reproducing, republishing or otherwise using site content without permission: a copyright clause, not an anti-scraping one. The wrappers fetch on the user's behalf; a published data release or a hosted mirror needs the owner's decision first. The site has no robots.txt (404, checked 2026-10-12), so no Crawl-delay is published; the runtime waits at least 1 s before every request, retries included. Payloads are not redistributed as release assets.
- Women's league: the W League is Japan's top women's basketball league (the SportsDataverse R counterpart is wehoop). All text is Japanese (team names, phases, play-by-play actions); the column descriptions translate the vocabulary.
- Seasons are END years (2026 = 2025-26, 2000 = 1999-2000). The API keys a season edition by an opaque league_id (45 = '25-26 W League'; every all-star game, cup and camp has one too) and the wrappers map the END year with the live league list (one extra, cached request; END = 2000 + the second YY of league_title). Phases inside an edition (season_id: 169 = regular season Premier, 170 = Future, 176 = playoff final; two divisions from 2024-25) are native ids, passed as given: wjbl_standing_seasons lists them with their END year.
- The schedule is paged by month: wjbl_schedule answers ONE month (game_month, YYYY-MM; omitted = the season's latest month only). List the months with wjbl_game_months and loop. The game routes take only the global play_schedule_id and serve final games only: each makes one extra, cached request (the game header) to prove the game is final and that the answer is the game and the teams asked for.

- Errors are HTTP 200: an unknown id answers [] (wehoop_no_data) and a missing or non-numeric parameter answers an error envelope (wehoop_asset_fetch). Coverage: schedules from 1999-2000, box scores from 2013-14, play-by-play and the period stats from 2018-19 (earlier games answer [] on the /pbp routes: wehoop_no_data). No route carries shot coordinates.
- Personal data: the organiser's contact details (phones, e-mail) and the player's blood type are never output, and phone numbers and e-mail addresses inside free text are replaced by [redacted-phone] / [redacted-email]. Do not add them back. Player bio fields and referee names are public and stay.
- Errors are classed conditions, all inheriting `wehoop_error`: `wehoop_no_data` (HTTP 404 or the API has nothing for the request: an empty list, an id it does not know, a game without play-by-play before 2018-19, a season without months or games), `wehoop_bad_request` (a season that is not the END year of a W League edition, an id that is not an integer or a game_month that is not YYYY-MM, raised before the request, or the API's HTTP 400 / 422) and `wehoop_asset_fetch` (any other failure: an error envelope at HTTP 200, any other non-2xx, an empty or non-JSON body, a game that is not final, a truncated play-by-play, or a body for another game, league edition, month, team, season phase or player than the one asked for; `transient` is TRUE for 408 / 429 / 5xx and transport failures). HTTP 500 is retried. Wrappers raise; they never warn and return an empty frame.

```
wjbl_teams()
```

## Value

A `wehoop_data` tibble, one row per record; the columns are always these, in this order (sdv-py `wjbl_teams`):

| col_name | type | description |
|---|---|---|
| team_id | integer | Franchise id team_id (key space: W League franchises, STABLE across seasons and sponsor renam |
| team_name | character | Franchise name, Japanese or English (Fujitsu Red Wave; all-star side Team Sirius). |

## See Also

Other WJBL Functions: `wjbl_box_score()`, `wjbl_game()`, `wjbl_game_info()`, `wjbl_game_months()`, `wjbl_head_to_head()`, `wjbl_leagues()`, `wjbl_pbp()`, `wjbl_period_scores()`, `wjbl_player()`, `wjbl_player_period_leaders()`, `wjbl_roster()`, `wjbl_schedule()`, `wjbl_staff()`, `wjbl_standing_seasons()`, `wjbl_team_box_score()`, `wjbl_team_period_stats()`, `wjbl_team_ranking()`, `wjbl_team_standings()`

---

`wnba_alltimeleadersgrids`

# Get WNBA Stats API All-time Leaders Grid

---

## Description

**Get WNBA Stats API All-time Leaders Grid**

**Get WNBA Stats API All-time Leaders Grid**

## Usage

```
wnba_alltimeleadersgrids(
  league_id = "10",
  per_mode = "PerGame",
  season_type = "Regular Season",
  top_x = 10,
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| per_mode | Per Mode - PerGame, Totals |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| top_x | Top X |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_alltimeleadersgrids(league_id = '10')
```

## Value

Returns a named list of data frames: ASTLeaders, BLKLeaders, DREBLeaders, FG3ALeaders, FG3MLeaders, FG3_PCTLeaders, FGALeaders, FGMLeaders, FG_PCTLeaders, FTALeaders, FTMLeaders, FT_PCTLeaders, GPLeaders, OREBLeaders, PFLeaders, PTSLeaders, REBLeaders, STLLeaders, TOVLeaders

**GPLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| GP | character | Games played. |
| GP_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**PTSLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| PTS | character | Points scored. |
| PTS_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**ASTLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| AST | character | Assists. |
| AST_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**STLLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| STL | character | Steals. |
| STL_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**OREBLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| OREB | character | Offensive rebounds. |
| OREB_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**DREBLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| DREB | character | Defensive rebounds. |
| DREB_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**REBLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| REB | character | Total rebounds. |
| REB_RANK | character | |

| | |
|---|---|
| IS_ACTIVE_FLAG | character |

**BLKLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| BLK | character | Blocks. |
| BLK_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FGMLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FGM | character | Field goals made. |
| FGM_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FGALeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FGA | character | Field goal attempts. |
| FGA_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FG_PCTLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG_PCT_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**TOVLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |

| | | |
|---|---|---|
| PLAYER_NAME | character | Player name. |
| TOV | character | Turnovers. |
| TOV_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FG3MLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FG3M | character | Three-point field goals made. |
| FG3M_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FG3ALeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FG3A | character | Three-point field goal attempts. |
| FG3A_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FG3_PCTLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FG3_PCT_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**PFLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| PF | character | Personal fouls. |
| PF_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FTMLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FTM | character | Free throws made. |
| FTM_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FTALeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FTA | character | Free throw attempts. |
| FTA_RANK | character | |
| IS_ACTIVE_FLAG | character | |

**FT_PCTLeaders**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FT_PCT | character | Free throw percentage (0-1). |
| FT_PCT_RANK | character | |
| IS_ACTIVE_FLAG | character | |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Leaders Functions: `wnba_assistleaders()`, `wnba_assisttracker()`, `wnba_homepagewidget()`, `wnba_leagueleaders()`

---

`wnba_assistleaders` **Get WNBA Stats API Assist Leaders**

---

## Description

**Get WNBA Stats API Assist Leaders**

**Get WNBA Stats API Assist Leaders**

## Usage

```
wnba_assistleaders(
  league_id = "10",
  per_mode = "PerGame",
  player_or_team = "Team",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| per_mode | Per Mode - PerGame, Totals |
| player_or_team | Player or Team |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_assistleaders(league_id = '10', player_or_team = "Player")
wnba_assistleaders(league_id = '10', player_or_team = "Team")
```

## Value

Returns a named list of data frames: AssistLeaders

**AssistLeaders**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Leaders Functions: wnba_alltimeleadersgrids(), wnba_assisttracker(), wnba_homepagewidget(), wnba_leagueleaders()

`wnba_assisttracker` **Get WNBA Stats API Assist Tracker**

## Description

**Get WNBA Stats API Assist Tracker**

**Get WNBA Stats API Assist Tracker**

## Usage

```
wnba_assisttracker(
  league_id = "10",
  per_mode = "PerGame",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `per_mode` | Per Mode - PerGame, Totals |
| `season` | Season - format 2020-21 |
| `season_type` | Season Type - Regular Season, Playoffs, All-Star |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_assisttracker(league_id = '10')
```

## Value

Returns a named list of data frames: AssistTracker

**AssistTracker**

| col_name | types | description |
|---|---|---|
| ASSISTS | numeric | Total assists. |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Leaders Functions: `wnba_alltimeleadersgrids()`, `wnba_assistleaders()`, `wnba_homepagewidget()`, `wnba_leagueleaders()`

wnba_boxscoresummaryv3

# Get WNBA Stats API Boxscore Summary V3

## Description

**Get WNBA Stats API Boxscore Summary V3**

**Get WNBA Stats API Boxscore Summary V3**

## Usage

```
wnba_boxscoresummaryv3(game_id = "1022200034", ...)
```

## Arguments

| | |
|---|---|
| game_id | Game ID - 10-digit zero-padded ID (e.g., '1022200034') |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_boxscoresummaryv3(game_id = "1022200034")
```

## Value

Returns a named list of data frames: game_summary, game_info, arena_info, officials, line_score, inactive_players, last_five_meetings, other_stats, available_video

**game_summary**

See the HTML help or pkgdown reference for the column table.

**line_score**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Boxscore V3 Functions: wnba_boxscorehustlev2(), wnba_boxscoretraditionalv3()

wnba_boxscoretraditionalv2

## Get WNBA Stats API Boxscore Traditional V2

### Description

**Get WNBA Stats API Boxscore Traditional V2**

**Get WNBA Stats API Boxscore Traditional V2**

**Get WNBA Stats API Boxscore Advanced V2**

**Get WNBA Stats API Boxscore Four Factors V2**

**Get WNBA Stats API Boxscore Misc V2**

**Get WNBA Stats API Boxscore Scoring V2**

**Get WNBA Stats API Boxscore Usage V2**

### Usage

```
wnba_boxscoretraditionalv2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscoreadvancedv2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscorefourfactorsv2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)
```

```
wnba_boxscoremiscv2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscorescoringv2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscoreusagev2(
  game_id,
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)
```

## Arguments

| | |
|---|---|
| game_id | Game ID |
| start_period | start_period |
| end_period | end_period |
| start_range | start_range |
| end_range | end_range |
| range_type | range_type |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_boxscoretraditionalv2(game_id = "1022200034")

wnba_boxscoreadvancedv2(game_id = "1022200034")
```

```
wnba_boxscorefourfactorsv2(game_id = "1022200034")

wnba_boxscoremiscv2(game_id = "1022200034")

wnba_boxscorescoringv2(game_id = "1022200034")

wnba_boxscoreusagev2(game_id = "1022200034")
```

## Value

A list of data frames: PlayerStats, TeamStarterBenchStats, TeamStats

**PlayerStats**

See the HTML help or pkgdown reference for the column table.

**TeamStats**

See the HTML help or pkgdown reference for the column table.

**TeamStarterBenchStats**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: PlayerStats, TeamStats

**PlayerStats**

See the HTML help or pkgdown reference for the column table.

**TeamStats**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: sqlPlayersFourFactors, sqlTeamFourFactors

**sqlPlayersFourFactors**

See the HTML help or pkgdown reference for the column table.

**sqlTeamsFourFactors**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: sqlPlayersMisc, sqlTeamsMisc

**sqlPlayersMisc**

See the HTML help or pkgdown reference for the column table.

**sqlTeamsMisc**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: sqlPlayersScoring, sqlTeamsScoring

**sqlPlayersScoring**

See the HTML help or pkgdown reference for the column table.

**sqlTeamsScoring**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: sqlPlayersUsage, sqlTeamsUsage

**sqlPlayersUsage**

See the HTML help or pkgdown reference for the column table.

**sqlTeamsUsage**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Boxscore Functions: wnba_boxscoresummaryv2(), wnba_gamerotation()

---

wnba_boxscoretraditionalv3

**Get WNBA Stats API Boxscore Traditional V3**

---

## Description

**Get WNBA Stats API Boxscore Traditional V3**

**Get WNBA Stats API Boxscore Traditional V3**

**Get WNBA Stats API Boxscore Advanced V3**

**Get WNBA Stats API Boxscore Misc V3**

**Get WNBA Stats API Boxscore Scoring V3**

**Get WNBA Stats API Boxscore Four Factors V3**

**Get WNBA Stats API Boxscore Player Tracking V3**

**Get WNBA Stats API Boxscore Usage V3**

## Usage

```
wnba_boxscoretraditionalv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscoreadvancedv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)
```

```
wnba_boxscoremiscv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscorescoringv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscorefourfactorsv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscoreplayertrackv3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
)

wnba_boxscoreusagev3(
  game_id = "1022200034",
  start_period = 0,
  end_period = 14,
  start_range = 0,
  end_range = 0,
  range_type = 0,
  ...
```

)

## Arguments

| | |
|---|---|
| game_id | Game ID - 10-digit zero-padded ID (e.g., '1022200034') |
| start_period | start_period |
| end_period | end_period |
| start_range | start_range |
| end_range | end_range |
| range_type | range_type |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_boxscoretraditionalv3(game_id = "1022200034")

wnba_boxscoreadvancedv3(game_id = "1022200034")

wnba_boxscoremiscv3(game_id = "1022200034")

wnba_boxscorescoringv3(game_id = "1022200034")

wnba_boxscorefourfactorsv3(game_id = "1022200034")

wnba_boxscoreplayertrackv3(game_id = "1022200034")

wnba_boxscoreusagev3(game_id = "1022200034")
```

## Value

A list of data frames: home_team_player_traditional, away_team_player_traditional, home_team_totals_traditional, away_team_totals_traditional, home_team_starters_totals, away_team_starters_totals, home_team_bench_totals, away_team_bench_totals

**home_team_player_traditional**

See the HTML help or pkgdown reference for the column table.

**away_team_player_traditional**

Same columns as the **home_team_player_traditional** table above.

**home_team_totals_traditional**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_traditional**

Same columns as the **home_team_totals_traditional** table above.

**home_team_starters_totals**

See the HTML help or pkgdown reference for the column table.

**away_team_starters_totals**

Same columns as the **home_team_starters_totals** table above.

**home_team_bench_totals**

Same columns as the **home_team_starters_totals** table above.

**away_team_bench_totals**

Same columns as the **home_team_starters_totals** table above.

A list of data frames: home_team_player_advanced, away_team_player_advanced, home_team_totals_advanced, away_team_totals_advanced

**home_team_player_advanced**

See the HTML help or pkgdown reference for the column table.

**away_team_player_advanced**

Same columns as the **home_team_player_advanced** table above.

**home_team_totals_advanced**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_advanced**

Same columns as the **home_team_totals_advanced** table above.

A list of data frames: home_team_player_misc, away_team_player_misc, home_team_totals_misc, away_team_totals_misc

**home_team_player_misc**

See the HTML help or pkgdown reference for the column table.

**away_team_player_misc**

Same columns as the **home_team_player_misc** table above.

**home_team_totals_misc**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_misc**

Same columns as the **home_team_totals_misc** table above.

A list of data frames: home_team_player_scoring, away_team_player_scoring, home_team_totals_scoring, away_team_totals_scoring

**home_team_player_scoring**

See the HTML help or pkgdown reference for the column table.

**away_team_player_scoring**

Same columns as the **home_team_player_scoring** table above.

**home_team_totals_scoring**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_scoring**

Same columns as the **home_team_totals_scoring** table above.

A list of data frames: home_team_player_four_factors, away_team_player_four_factors, home_team_totals_four_factors, away_team_totals_four_factors

**home_team_player_four_factors**

See the HTML help or pkgdown reference for the column table.

**away_team_player_four_factors**

Same columns as the **home_team_player_four_factors** table above.

**home_team_totals_four_factors**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_four_factors**

Same columns as the **home_team_totals_four_factors** table above.

A list of data frames: home_team_player_player_track, away_team_player_player_track, home_team_totals_player_track, away_team_totals_player_track

**home_team_player_player_track**

See the HTML help or pkgdown reference for the column table.

**away_team_player_player_track**

Same columns as the **home_team_player_player_track** table above.

**home_team_totals_player_track**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_player_track**

Same columns as the **home_team_totals_player_track** table above.

Returns a named list of data frames: home_team_player_usage, away_team_player_usage, home_team_totals_usage, away_team_totals_usage

**home_team_player_usage**

See the HTML help or pkgdown reference for the column table.

**away_team_player_usage**

Same columns as the **home_team_player_usage** table above.

**home_team_totals_usage**

See the HTML help or pkgdown reference for the column table.

**away_team_totals_usage**

Same columns as the **home_team_totals_usage** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Boxscore V3 Functions: `wnba_boxscorehustlev2()`, `wnba_boxscoresummaryv3()`

`wnba_commonallplayers` **Get WNBA Stats API All Players**

## Description

**Get WNBA Stats API All Players**

**Get WNBA Stats API All Players**

## Usage

```
wnba_commonallplayers(
  is_only_current_season = 0,
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  ...
)
```

## Arguments

| | |
|---|---|
| `is_only_current_season` | is_only_current_season |
| `league_id` | league_id |
| `season` | season |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_commonallplayers(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: CommonAllPlayers

**CommonAllPlayers**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Roster Functions: `wnba_commonplayerinfo()`, `wnba_commonteamroster()`

`wnba_commonplayerinfo` **Get WNBA Stats API Player Info**

## Description

**Get WNBA Stats API Player Info**

**Get WNBA Stats API Player Info**

## Usage

```
wnba_commonplayerinfo(league_id = "10", player_id = "1628932", ...)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `player_id` | player_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_commonplayerinfo(league_id = '10', player_id = '1628932')
```

## Value

Return a named list of data frames: CommonPlayerInfo, PlayerHeadlineStats, AvailableSeasons

**CommonPlayerInfo**

See the HTML help or pkgdown reference for the column table.

**PlayerHeadlineStats**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| TimeFrame | character | |
| PTS | character | Points scored. |
| AST | character | Assists. |
| REB | character | Total rebounds. |
| ALL_STAR_APPEARANCES | character | |

**AvailableSeasons**

| col_name | types | description |
|---|---|---|
| SEASON_ID | character | Unique season identifier. |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Roster Functions: wnba_commonallplayers(), wnba_commonteamroster()

---

wnba_commonplayoffseries

**Get WNBA Stats API Playoff Series**

---

## Description

**Get WNBA Stats API Playoff Series**

**Get WNBA Stats API Playoff Series**

## Usage

```
wnba_commonplayoffseries(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  series_id = "",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | league_id |
| season | season |
| series_id | series_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_commonplayoffseries(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: PlayoffSeries

**PlayoffSeries**

| col_name | types | description |
|---|---|---|
| GAME_ID | character | Unique game identifier. |
| HOME_TEAM_ID | character | Unique identifier for the home team. |
| VISITOR_TEAM_ID | character | Unique identifier for visitor team. |

| | |
|---|---|
| SERIES_ID | character |
| GAME_NUM | character |

## Author(s)

Saiem Gilani

---

`wnba_commonteamroster` **Get WNBA Stats API Team Roster**

---

## Description

**Get WNBA Stats API Team Roster**

**Get WNBA Stats API Team Roster**

## Usage

```
wnba_commonteamroster(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  team_id = "1611661317",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `season` | season |
| `team_id` | team_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_commonteamroster(season = most_recent_wnba_season() - 1, team_id = '1611661317')
```

## Value

Return a named list of data frames: Coaches, CommonTeamRoster

**CommonTeamRoster**

See the HTML help or pkgdown reference for the column table.

**Coaches**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Roster Functions: wnba_commonallplayers(), wnba_commonplayerinfo()

---

wnba_cumestatsplayer **Get WNBA Stats API Cumulative Player Stats**

---

### Description

**Get WNBA Stats API Cumulative Player Stats**

**Get WNBA Stats API Cumulative Player Stats**

### Usage

```
wnba_cumestatsplayer(
  game_ids = "1022200018",
  league_id = "10",
  player_id = "204319",
  season = "2021-22",
  season_type = "Regular Season",
  team_id = "",
  ...
)
```

### Arguments

| | |
|---|---|
| game_ids | game_ids |
| league_id | league_id |
| player_id | player_id |
| season | season |
| season_type | season_type |
| team_id | team_id |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_cumestatsplayer(game_ids = "1022200018", player_id = "204319", season = "2021-22")
```

## Value

Return a named list of data frames: GameByGameStats, TotalPlayerStats

**GameByGameStats**

See the HTML help or pkgdown reference for the column table.

**TotalPlayerStats**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Cume Functions: `wnba_cumestatsplayergames()`, `wnba_cumestatsteam()`, `wnba_cumestatsteamgames()`

---

`wnba_cumestatsplayergames`

# Get WNBA Stats API Cumulative Player Game Stats

---

## Description

**Get WNBA Stats API Cumulative Player Game Stats**

**Get WNBA Stats API Cumulative Player Game Stats**

## Usage

```
wnba_cumestatsplayergames(
  league_id = "10",
  location = "",
  outcome = "",
  player_id = "204319",
  season = "2021-22",
  season_type = "Regular Season",
  vs_conference = "",
  vs_division = "",
  vs_team_id = "",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `location` | location |
| `outcome` | outcome |

| | |
|---|---|
| player_id | player_id |
| season | season |
| season_type | season_type |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_team_id | vs_team_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_cumestatsplayergames(player_id = "204319", season = "2021-22")
```

## Value

Returns a named list of data frames: CumeStatsPlayerGames

**CumeStatsPlayerGames**

| col_name | types | description |
|---|---|---|
| MATCHUP | character | Matchup. |
| GAME_ID | character | Unique game identifier. |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Cume Functions: wnba_cumestatsplayer(), wnba_cumestatsteam(), wnba_cumestatsteamgames()

---

# wnba_cumestatsteam Get WNBA Stats API Cumulative Team Stats

---

## Description

**Get WNBA Stats API Cumulative Team Stats**

**Get WNBA Stats API Cumulative Team Stats**

## Usage

```
wnba_cumestatsteam(
  game_ids = "1022200018",
  league_id = "10",
  season = "2021-22",
  season_type = "Regular Season",
  team_id = "1611661317",
  ...
)
```

### Arguments

| | |
|---|---|
| game_ids | game_ids |
| league_id | league_id |
| season | season |
| season_type | season_type |
| team_id | team_id |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_cumestatsteam(game_ids = "1022200018", season = "2021-22", team_id = "1611661317")
```

### Value

Returns a named list of data frames: GameByGameStats, TotalTeamStats

**GameByGameStats**

See the HTML help or pkgdown reference for the column table.

**TotalTeamStats**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Cume Functions: wnba_cumestatsplayer(), wnba_cumestatsplayergames(), wnba_cumestatsteamgames

---

## wnba_cumestatsteamgames

## Get WNBA Stats API Cumulative Team Game Stats

---

### Description

**Get WNBA Stats API Cumulative Team Game Stats**

**Get WNBA Stats API Cumulative Team Game Stats**

## Usage

```
wnba_cumestatsteamgames(
  league_id = "10",
  location = "",
  outcome = "",
  season = "2021-22",
  season_id = "",
  season_type = "Regular Season",
  team_id = 1611661317,
  vs_conference = "",
  vs_division = "",
  vs_team_id = "",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | league_id |
| location | location |
| outcome | outcome |
| season | season |
| season_id | season_id |
| season_type | season_type |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_team_id | vs_team_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
  wnba_cumestatsteamgames(team_id = 1611661317, season = "2021-22")
``
```

## Value

Returns a named list of data frames: CumeStatsTeamGames

**CumeStatsTeamGames**

| col_name | types | description |
|---|---|---|
| MATCHUP | character | Matchup. |
| GAME_ID | character | Unique game identifier. |

### Author(s)

Saiem Gilani

### See Also

Other WNBA Cume Functions: wnba_cumestatsplayer(), wnba_cumestatsplayergames(), wnba_cumestatsteam()

---

wnba_data_pbp **Get WNBA Data API Play-by-Play**

---

### Description

**Get WNBA Data API Play-by-Play**

**Get WNBA Data API Play-by-Play**

### Usage

```
wnba_data_pbp(game_id = "1022200034", ...)
```

### Arguments

| | |
|---|---|
| game_id | Game ID - 10 digits, i.e. "0021900001" |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_data_pbp(game_id = "1022200034")
```

### Value

Returns a tibble

See the HTML help or pkgdown reference for the column table.

Event Message Types (event_type):

1 -> MAKE

2 -> MISS

3 -> FreeThrow

4 -> Rebound

5 -> Turnover

6 -> Foul

7 -> Violation

8 -> Substitution

9 -> Timeout

10 -> JumpBall

11 -> Ejection

12 -> StartOfPeriod

13 -> EndOfPeriod

14 -> Empty

## Author(s)

Saiem Gilani

## See Also

Other WNBA PBP Functions: `wnba_live_pbp()`, `wnba_pbp()`, `wnba_pbps()`, `wnba_playbyplayv3()`

---

`wnba_draftboard` **Get WNBA Stats API Draft Board**

---

## Description

**Get WNBA Stats API Draft Board**

**Get WNBA Stats API Draft Board**

## Usage

```
wnba_draftboard(season = most_recent_wnba_season(), ...)
```

## Arguments

| | |
|---|---|
| `season` | Draft year as numeric or character (e.g. `2026`). |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_draftboard(season = most_recent_wnba_season())
```

## Value

Returns a named list of tibbles: `board`, `picks`.

**board**

| col_name | types | description |
|---|---|---|
| draft_id | integer | Unique identifier for draft. |
| title | character | Title or label for the record. |
| season | integer | Season identifier (4-digit year or 'YYYY-YY' string). |
| status | character | Status label. |
| on_the_clock | character | On the clock. |
| draft_date | character | Date in YYYY-MM-DD format. |

| | | |
|---|---|---|
| modified | character | Modified. |

**picks**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Draft Functions: `wnba_draftcombinestats()`, `wnba_drafthistory()`

---

`wnba_draftcombinestats`

**Get WNBA Stats API Draft Combine Stats**

---

## Description

**Get WNBA Stats API Draft Combine Stats**

**Get WNBA Stats API Draft Combine Stats**

**Get WNBA Stats API Draft Combine Drill Results**

**Get WNBA Stats API Draft Combine Non-Stationary Shooting**

**Get WNBA Stats API Draft Combine Player Anthropological Measurements**

**Get WNBA Stats API Draft Combine - Spot Shooting**

## Usage

```
wnba_draftcombinestats(
  league_id = "10",
  season_year = most_recent_wnba_season() - 1,
  ...
)

wnba_draftcombinedrillresults(
  league_id = "10",
  season_year = most_recent_wnba_season() - 1,
  ...
)

wnba_draftcombinenonstationaryshooting(
  league_id = "10",
  season_year = most_recent_wnba_season() - 1,
  ...
```

```
)

wnba_draftcombineplayeranthro(
  league_id = "10",
  season_year = most_recent_wnba_season() - 1,
  ...
)

wnba_draftcombinespotshooting(
  league_id = "10",
  season_year = most_recent_wnba_season() - 1,
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | league_id |
| season_year | season_year |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_draftcombinestats(season_year = most_recent_wnba_season() - 1)
```

(Possibly Defunct)

```
wnba_draftcombinedrillresults(season_year = most_recent_wnba_season() - 2)
```

(Possibly Defunct)

```
wnba_draftcombinenonstationaryshooting(season_year = most_recent_wnba_season() - 2)
```

(Possibly Defunct)

```
wnba_draftcombineplayeranthro(season_year = most_recent_wnba_season() - 2)
```

(Possibly Defunct)

```
wnba_draftcombinespotshooting(season_year = most_recent_wnba_season() - 2)
```

## Value

Returns a named list of data frames: DraftCombineStats

**DraftCombineStats**

See the HTML help or pkgdown reference for the column table.

Returns a named list of data frames: Results

Returns a named list of data frames: Results

Returns a named list of data frames: Results

Returns a named list of data frames: Results

### Author(s)

Saiem Gilani

### See Also

Other WNBA Draft Functions: `wnba_draftboard()`, `wnba_drafthistory()`

---

`wnba_drafthistory` **Get WNBA Stats API Draft History**

---

### Description

**Get WNBA Stats API Draft History**

### Usage

```
wnba_drafthistory(
  league_id = "10",
  college = "",
  overall_pick = "",
  round_pick = "",
  round_num = "",
  season = most_recent_wnba_season() - 1,
  team_id = "",
  top_x = "",
  ...
)
```

### Arguments

| | |
|---|---|
| `league_id` | league_id |
| `college` | college |
| `overall_pick` | overall_pick |
| `round_pick` | round_pick |
| `round_num` | round_num |
| `season` | season |
| `team_id` | team_id |
| `top_x` | top_x |
| `...` | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_drafthistory(season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: DraftHistory

**DraftHistory**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Draft Functions: `wnba_draftboard()`, `wnba_draftcombinestats()`

---

`wnba_fantasywidget` **Get WNBA Stats API Fantasy Widget**

---

## Description

**Get WNBA Stats API Fantasy Widget**

**Get WNBA Stats API Fantasy Widget**

## Usage

```
wnba_fantasywidget(
  active_players = "N",
  date_from = "",
  date_to = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  month = "",
  opponent_team_id = "",
  po_round = "",
  player_id = "",
  position = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  team_id = "",
  todays_opponent = 0,
  todays_players = "N",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| active_players | active_players |
| date_from | date_from date_from |
| date_to | date_to date_to |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| month | month |
| opponent_team_id | opponent_team_id |
| po_round | po_round |
| player_id | player_id |
| position | position |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| team_id | team_id |
| todays_opponent | todays_opponent |
| todays_players | todays_players |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_fantasywidget(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: FantasyWidgetResult

**FantasyWidgetResult**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Fantasy Functions: `wnba_infographicfanduelplayer()`, `wnba_playerfantasyprofile()`, `wnba_playerfantasyprofilebargraph()`

`wnba_franchisehistory` **Get WNBA Stats API Franchise History**

## Description

**Get WNBA Stats API Franchise History**

**Get WNBA Stats API Franchise History**

## Usage

```
wnba_franchisehistory(league_id = "10", ...)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

Franchise History

```
wnba_franchisehistory(league_id = '10')
```

## Value

Returns a named list of data frames: DefunctTeams, FranchiseHistory

**FranchiseHistory**

See the HTML help or pkgdown reference for the column table.

**DefunctTeams**

Same columns as the **FranchiseHistory** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Franchise Functions: `wnba_franchiseleaders()`, `wnba_franchiseleaderswrank()`, `wnba_franchiseplayers()`

`wnba_franchiseleaders` **Get WNBA Stats API Franchise Leaders**

## Description

**Get WNBA Stats API Franchise Leaders**

**Get WNBA Stats API Franchise Leaders**

## Usage

```
wnba_franchiseleaders(league_id = "10", team_id = "1611661324", ...)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `team_id` | team_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

Franchise Leaders

```
wnba_franchiseleaders(league_id = '10', team_id = '1611661324')
```

## Value

Returns a named list of data frames: FranchiseLeaders

**FranchiseLeaders**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Franchise Functions: `wnba_franchisehistory()`, `wnba_franchiseleaderswrank()`, `wnba_franchiseplayers()`

wnba_franchiseleaderswrank

## Get WNBA Stats API Franchise Leaders with Rank

### Description

**Get WNBA Stats API Franchise Leaders with Rank**

**Get WNBA Stats API Franchise Leaders with Rank**

### Usage

```
wnba_franchiseleaderswrank(
  league_id = "10",
  per_mode = "Totals",
  season_type = "Regular Season",
  team_id = "1611661324",
  ...
)
```

### Arguments

| | |
|---|---|
| league_id | league_id |
| per_mode | per_mode |
| season_type | season_type |
| team_id | team_id |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

Franchise Leaders

```
wnba_franchiseleaderswrank(league_id = '10', team_id = '1611661324')
```

### Value

Returns a named list of data frames: FranchiseLeaderswRank

**FranchiseLeaderswRank**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Franchise Functions: wnba_franchisehistory(), wnba_franchiseleaders(), wnba_franchiseplayers()

wnba_franchiseplayers **Get WNBA Stats API Franchise Players**

## Description

**Get WNBA Stats API Franchise Players**

**Get WNBA Stats API Franchise Players**

## Usage

```
wnba_franchiseplayers(
  league_id = "10",
  per_mode = "Totals",
  season_type = "Regular Season",
  team_id = "1611661319",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | league_id |
| per_mode | per_mode |
| season_type | season_type |
| team_id | team_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

Franchise Players

```
wnba_franchiseplayers(league_id = '10', team_id = '1611661319')
wnba_franchiseplayers(league_id = '10', season_type = 'Playoffs', team_id = '1611661319')
```

## Value

Returns a named list of data frames: FranchisePlayers

**FranchisePlayers**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Franchise Functions: wnba_franchisehistory(), wnba_franchiseleaders(), wnba_franchiseleaderswr

wnba_gamerotation **Get WNBA Stats API Game Rotation**

## Description

**Get WNBA Stats API Game Rotation**

**Get WNBA Stats API Game Rotation**

## Usage

```
wnba_gamerotation(game_id, league_id = "10", rotation_stat = "PLAYER_PTS", ...)
```

## Arguments

| | |
|---|---|
| game_id | Game ID |
| league_id | League ID |
| rotation_stat | Rotation stat to provide details on: PLAYER_PTS, PT_DIFF, USG_PCT |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_gamerotation(game_id = "1022200034")
```

## Value

Returns a named list of data frames: AwayTeam, HomeTeam

**AwayTeam**

See the HTML help or pkgdown reference for the column table.

**HomeTeam**

Same columns as the **AwayTeam** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Boxscore Functions: wnba_boxscoresummaryv2(), wnba_boxscoretraditionalv2()

Other WNBA Lineup Functions: wnba_leaguedashlineups(), wnba_leaguelineupviz(), wnba_possession_lineups()
wnba_rapm(), wnba_shot_zones(), wnba_shotchartlineupdetail(), wnba_teamdashlineups()

`wnba_homepagewidget` **Get WNBA Stats API Homepage Widget Leaders**

## Description

**Get WNBA Stats API Homepage Widget Leaders**

**Get WNBA Stats API Homepage Widget Leaders**

## Usage

```
wnba_homepagewidget(player_or_team = "Player", ...)
```

## Arguments

| | |
|---|---|
| `player_or_team` | Player or Team |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_homepagewidget(player_or_team = "Player")
wnba_homepagewidget(player_or_team = "Team")
```

## Value

Returns, for current season, a named list of data frames: PTS, REB, AST, BLK, STL, FG_PCT, FG3M, FG3_PCT, FANTASY_POINTS, TOTAL_PTS, TOTAL_REB, TOTAL_AST, ALL_TIME_TD3, TD3, GAME_FG3M, GAME_PTS, GAME_AST, GAME_STL, GAME_BLK, PCT_PTS_3PT, PCT_PTS_2PT, PCT_PTS_2PT_MR

**PTS**

See the HTML help or pkgdown reference for the column table.

**REB**

See the HTML help or pkgdown reference for the column table.

**AST**

See the HTML help or pkgdown reference for the column table.

**BLK**

See the HTML help or pkgdown reference for the column table.

**STL**

See the HTML help or pkgdown reference for the column table.

**FG_PCT**

See the HTML help or pkgdown reference for the column table.

**FG3M**

See the HTML help or pkgdown reference for the column table.

**FG3_PCT**

See the HTML help or pkgdown reference for the column table.

**FANTASY_POINTS**

See the HTML help or pkgdown reference for the column table.

**TOTAL_PTS**

See the HTML help or pkgdown reference for the column table.

**TOTAL_REB**

See the HTML help or pkgdown reference for the column table.

**TOTAL_AST**

See the HTML help or pkgdown reference for the column table.

**ALL_TIME_TD3**

See the HTML help or pkgdown reference for the column table.

**TD3**

See the HTML help or pkgdown reference for the column table.

**GAME_FG3M**

See the HTML help or pkgdown reference for the column table.

**GAME_PTS**

See the HTML help or pkgdown reference for the column table.

**GAME_AST**

See the HTML help or pkgdown reference for the column table.

**GAME_STL**

See the HTML help or pkgdown reference for the column table.

**GAME_BLK**

See the HTML help or pkgdown reference for the column table.

**PCT_PTS_3PT**

See the HTML help or pkgdown reference for the column table.

**PCT_PTS_2PT**

See the HTML help or pkgdown reference for the column table.

**PCT_PTS_2PT_MR**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Leaders Functions: `wnba_alltimeleadersgrids()`, `wnba_assistleaders()`, `wnba_assisttracker()`, `wnba_leagueleaders()`

wnba_infographicfanduelplayer

**Get WNBA Stats API FanDuel Player Infographic**

## Description

**Get WNBA Stats API FanDuel Player Infographic**

**Get WNBA Stats API FanDuel Player Infographic**

## Usage

```
wnba_infographicfanduelplayer(game_id, ...)
```

## Arguments

| | |
|---|---|
| game_id | game_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_infographicfanduelplayer(game_id = "1022200034")
```

## Value

Return a named list of data frames: FanDuelPlayer

**FanDuelPlayer**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Fantasy Functions: wnba_fantasywidget(), wnba_playerfantasyprofile(), wnba_playerfantasyprofil

wnba_leaguedashlineups

## Get WNBA Stats API League Dashboard Lineups

### Description

**Get WNBA Stats API League Dashboard Lineups**

**Get WNBA Stats API League Dashboard Lineups**

### Usage

```
wnba_leaguedashlineups(
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  game_segment = "",
  group_quantity = 5,
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

### Arguments

| | |
|---|---|
| conference | conference |
| date_from | date_from |
| date_to | date_to |

| | |
|---|---|
| division | division |
| game_segment | game_segment |
| group_quantity | group_quantity |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashlineups(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: Lineups

**Lineups**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Lineup Functions: wnba_gamerotation(), wnba_leaguelineupviz(), wnba_possession_lineups(), wnba_rapm(), wnba_shot_zones(), wnba_shotchartlineupdetail(), wnba_teamdashlineups()

---

# wnba_leaguedashplayerbiostats

**Get WNBA Stats API League Dashboard Player Biographical Stats**

---

## Description

**Get WNBA Stats API League Dashboard Player Biographical Stats**

**Get WNBA Stats API League Dashboard Player Biographical Stats**

## Usage

```
wnba_leaguedashplayerbiostats(
  college = "",
  conference = "",
  country = "",
  date_from = "",
  date_to = "",
  division = "",
  draft_pick = "",
  draft_year = "",
  game_segment = "",
  game_scope = "",
  height = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  per_mode = "Totals",
  period = "",
  player_experience = "",
  player_position = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
```

```
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
  touch_time_range = "",
  vs_conference = "",
  vs_division = "",
  weight = "",
  ...
)
```

## Arguments

| | |
|---|---|
| college | college |
| conference | conference |
| country | country |
| date_from | date_from |
| date_to | date_to |
| division | division |
| draft_pick | draft_pick |
| draft_year | draft_year |
| game_segment | game_segment |
| game_scope | game_scope |
| height | height |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_experience | player_experience |
| player_position | player_position |
| season | season |
| season_segment | season_segment |
| season_type | season_type |

shot_clock_range
: shot_clock_range

starter_bench
: starter_bench

team_id
: team_id

touch_time_range
: touch_time_range

vs_conference
: vs_conference

vs_division
: vs_division

weight
: weight

...
: Additional arguments passed to an underlying function like httr.

## Details

```
wnba_leaguedashplayerbiostats(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueDashPlayerBioStats

**LeagueDashPlayerBioStats**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll, wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet, wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

wnba_leaguedashplayerclutch

**Get WNBA Stats API League Dashboard by Player Clutch Splits**

## Description

**Get WNBA Stats API League Dashboard by Player Clutch Splits**

**Get WNBA Stats API League Dashboard by Player Clutch Splits**

## Usage

```
wnba_leaguedashplayerclutch(
  ahead_behind = "Ahead or Behind",
  clutch_time = "Last 5 Minutes",
  college = "",
  conference = "",
  country = "",
  date_from = "",
  date_to = "",
  division = "",
  draft_pick = "",
  draft_year = "",
  game_scope = "",
  game_segment = "",
  height = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  point_diff = 5,
  po_round = "",
  per_mode = "Totals",
  period = 0,
  player_experience = "",
  player_position = "",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
```

```
  team_id = "",
  touch_time_range = "",
  vs_conference = "",
  vs_division = "",
  weight = "",
  ...
)
```

## Arguments

| | |
|---|---|
| ahead_behind | ahead_behind |
| clutch_time | clutch_time |
| college | college |
| conference | conference |
| country | country |
| date_from | date_from |
| date_to | date_to |
| division | division |
| draft_pick | draft_pick |
| draft_year | draft_year |
| game_scope | game_scope |
| game_segment | game_segment |
| height | height |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| plus_minus | plus_minus |
| point_diff | point_diff |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_experience | player_experience |
| player_position | player_position |

| | |
|---|---|
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| starter_bench | starter_bench |
| team_id | team_id |
| touch_time_range | touch_time_range |
| vs_conference | vs_conference |
| vs_division | vs_division |
| weight | weight |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashplayerclutch(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueDashPlayerClutch

**LeagueDashPlayerClutch**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll, wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet, wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

wnba_leaguedashplayershotlocations

**Get WNBA Stats API League Dashboard Player Shot Locations**

## Description

**Get WNBA Stats API League Dashboard Player Shot Locations**

**Get WNBA Stats API League Dashboard Player Shot Locations**

## Usage

```
wnba_leaguedashplayershotlocations(
  college = "",
  conference = "",
  country = "",
  date_from = "",
  date_to = "",
  distance_range = "By Zone",
  division = "",
  draft_pick = "",
  draft_year = "",
  dribble_range = "",
  game_scope = "",
  game_segment = "",
  height = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_experience = "",
  player_position = "",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
```

```
  vs_conference = "",
  vs_division = "",
  weight = "",
  ...
)
```

## Arguments

| | |
|---|---|
| college | college |
| conference | conference |
| country | country |
| date_from | date_from |
| date_to | date_to |
| distance_range | distance_range |
| division | division |
| draft_pick | draft_pick |
| draft_year | draft_year |
| dribble_range | dribble_range |
| game_scope | game_scope |
| game_segment | game_segment |
| height | height |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| player_experience | player_experience |
| player_position | player_position |
| plus_minus | plus_minus |
| rank | rank |
| season | season |

| | |
|---|---|
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| starter_bench | starter_bench |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| weight | weight |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashplayershotlocations(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: ShotLocations

**ShotLocations**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll, wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet, wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Shooting Functions: wnba_leaguedashteamshotlocations(), wnba_playerdashboardbyclutch(), wnba_shotchartdetail(), wnba_shotchartleaguewide(), wnba_shotchartlineupdetail(), wnba_teamdashboardbyclutch()

wnba_leaguedashplayerstats

**Get WNBA Stats API League Dashboard Player Stats**

## Description

**Get WNBA Stats API League Dashboard Player Stats**

**Get WNBA Stats API League Dashboard Player Stats**

## Usage

```
wnba_leaguedashplayerstats(
  college = "",
  conference = "",
  country = "",
  date_from = "",
  date_to = "",
  division = "",
  draft_pick = "",
  draft_year = "",
  game_scope = "",
  game_segment = "",
  height = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  player_experience = "",
  player_position = "",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
  two_way = "",
  vs_conference = "",
```

```
  vs_division = "",
  weight = "",
  ...
)
```

## Arguments

| | |
|---|---|
| college | college |
| conference | conference |
| country | country |
| date_from | date_from |
| date_to | date_to |
| division | division |
| draft_pick | draft_pick |
| draft_year | draft_year |
| game_scope | game_scope |
| game_segment | game_segment |
| height | height |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_experience | player_experience |
| player_position | player_position |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |

| | |
|---|---|
| starter_bench | starter_bench |
| team_id | team_id |
| two_way | two_way |
| vs_conference | vs_conference |
| vs_division | vs_division |
| weight | weight |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashplayerstats(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueDashPlayerStats

**LeagueDashPlayerStats**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll, wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet, wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

wnba_leaguedashteamclutch

**Get WNBA Stats API League Dashboard by Team Clutch Splits**

## Description

**Get WNBA Stats API League Dashboard by Team Clutch Splits**

**Get WNBA Stats API League Dashboard by Team Clutch Splits**

## Usage

```
wnba_leaguedashteamclutch(
  ahead_behind = "Ahead or Behind",
  clutch_time = "Last 5 Minutes",
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  game_scope = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  point_diff = 5,
  po_round = "",
  per_mode = "Totals",
  period = 0,
  player_experience = "",
  player_position = "",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| ahead_behind | ahead_behind |
| clutch_time | clutch_time |
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| division | division |
| game_scope | game_scope |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| plus_minus | plus_minus |
| point_diff | point_diff |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_experience | player_experience |
| player_position | player_position |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| starter_bench | starter_bench |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashteamclutch(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueDashTeamClutch

**LeagueDashTeamClutch**

Columns as documented in the shared wnba_leaguedashteamclutch_league_dash_team_clutch_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Clutch Functions: wnba_playerdashboardbyclutch(), wnba_teamdashboardbyclutch()

---

`wnba_leaguedashteamshotlocations`

**Get WNBA Stats API League Dashboard Team Shot Locations**

---

## Description

**Get WNBA Stats API League Dashboard Team Shot Locations**

**Get WNBA Stats API League Dashboard Team Shot Locations**

## Usage

```
wnba_leaguedashteamshotlocations(
  conference = "",
  date_from = "",
  date_to = "",
  distance_range = "By Zone",
  division = "",
  game_scope = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
```

```
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_experience = "",
  player_position = "",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| distance_range | distance_range |
| division | division |
| game_scope | game_scope |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |

| | |
|---|---|
| player_experience | player_experience |
| player_position | player_position |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| starter_bench | starter_bench |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashteamshotlocations(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: ShotLocations

**ShotLocations**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Shooting Functions: wnba_leaguedashplayershotlocations(), wnba_playerdashboardbyclutch(), wnba_shotchartdetail(), wnba_shotchartleaguewide(), wnba_shotchartlineupdetail(), wnba_teamdashboardbyclutch()

wnba_leaguedashteamstats

**Get WNBA Stats API League Dashboard Team Stats**

## Description

**Get WNBA Stats API League Dashboard Team Stats**

**Get WNBA Stats API League Dashboard Team Stats**

## Usage

```
wnba_leaguedashteamstats(
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  game_scope = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  starter_bench = "",
  team_id = "",
  two_way = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| conference | conference |

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| division | division |
| game_scope | game_scope |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| starter_bench | starter_bench |
| team_id | team_id |
| two_way | two_way |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguedashteamstats(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueDashTeamStats

**LeagueDashTeamStats**

Columns as documented in the shared wnba_leaguedashteamclutch_league_dash_team_clutch_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Team Functions: wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

# wnba_leaguegamefinder Get WNBA Stats API League Game Streak Finder

---

## Description

**Get WNBA Stats API League Game Streak Finder**

**Get WNBA Stats API League Game Streak Finder**

## Usage

```
wnba_leaguegamefinder(
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  draft_year = "",
  draft_team_id = "",
  draft_round = "",
  draft_number = "",
  et_ast = "",
  et_blk = "",
  et_dd = "",
  et_dreb = "",
  et_fg3a = "",
  et_fg3m = "",
  et_fg3_pct = "",
  et_fga = "",
  et_fgm = "",
  et_fg_pct = "",
  et_fta = "",
  et_ftm = "",
```

```
    et_ft_pct = "",
    et_minutes = "",
    et_oreb = "",
    et_pf = "",
    et_pts = "",
    et_reb = "",
    et_stl = "",
    et_td = "",
    et_tov = "",
    game_id = "",
    gt_ast = "",
    gt_blk = "",
    gt_dd = "",
    gt_dreb = "",
    gt_fg3a = "",
    gt_fg3m = "",
    gt_fg3_pct = "",
    gt_fga = "",
    gt_fgm = "",
    gt_fg_pct = "",
    gt_fta = "",
    gt_ftm = "",
    gt_ft_pct = "",
    gt_minutes = "",
    gt_oreb = "",
    gt_pf = "",
    gt_pts = "",
    gt_reb = "",
    gt_stl = "",
    gt_td = "",
    gt_tov = "",
    league_id = "10",
    location = "",
    lt_ast = "",
    lt_blk = "",
    lt_dd = "",
    lt_dreb = "",
    lt_fg3a = "",
    lt_fg3m = "",
    lt_fg3_pct = "",
    lt_fga = "",
    lt_fgm = "",
    lt_fg_pct = "",
    lt_fta = "",
    lt_ftm = "",
    lt_ft_pct = "",
    lt_minutes = "",
    lt_oreb = "",
```

```
  lt_pf = "",
  lt_pts = "",
  lt_reb = "",
  lt_stl = "",
  lt_td = "",
  lt_tov = "",
  outcome = "",
  po_round = "",
  player_id = "",
  player_or_team = "T",
  rookie_year = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  starter_bench = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  vs_team_id = "",
  years_experience = "",
  ...
)
```

## Arguments

| | |
|---|---|
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| division | division |
| draft_year | draft_year |
| draft_team_id | draft_team_id |
| draft_round | draft_round |
| draft_number | draft_number |
| et_ast | et_ast |
| et_blk | et_blk |
| et_dd | et_dd |
| et_dreb | et_dreb |
| et_fg3a | et_fg3a |
| et_fg3m | et_fg3m |
| et_fg3_pct | et_fg3_pct |
| et_fga | et_fga |
| et_fgm | et_fgm |
| et_fg_pct | et_fg_pct |

| | |
|---|---|
| et_fta | et_fta |
| et_ftm | et_ftm |
| et_ft_pct | et_ft_pct |
| et_minutes | et_minutes |
| et_oreb | et_oreb |
| et_pf | et_pf |
| et_pts | et_pts |
| et_reb | et_reb |
| et_stl | et_stl |
| et_td | et_td |
| et_tov | et_tov |
| game_id | game_id |
| gt_ast | gt_ast |
| gt_blk | gt_blk |
| gt_dd | gt_dd |
| gt_dreb | gt_dreb |
| gt_fg3a | gt_fg3a |
| gt_fg3m | gt_fg3m |
| gt_fg3_pct | gt_fg3_pct |
| gt_fga | gt_fga |
| gt_fgm | gt_fgm |
| gt_fg_pct | gt_fg_pct |
| gt_fta | gt_fta |
| gt_ftm | gt_ftm |
| gt_ft_pct | gt_ft_pct |
| gt_minutes | gt_minutes |
| gt_oreb | gt_oreb |
| gt_pf | gt_pf |
| gt_pts | gt_pts |
| gt_reb | gt_reb |
| gt_stl | gt_stl |
| gt_td | gt_td |
| gt_tov | gt_tov |
| league_id | league_id |
| location | location |
| lt_ast | lt_ast |
| lt_blk | lt_blk |

| | |
|---|---|
| lt_dd | lt_dd |
| lt_dreb | lt_dreb |
| lt_fg3a | lt_fg3a |
| lt_fg3m | lt_fg3m |
| lt_fg3_pct | lt_fg3_pct |
| lt_fga | lt_fga |
| lt_fgm | lt_fgm |
| lt_fg_pct | lt_fg_pct |
| lt_fta | lt_fta |
| lt_ftm | lt_ftm |
| lt_ft_pct | lt_ft_pct |
| lt_minutes | lt_minutes |
| lt_oreb | lt_oreb |
| lt_pf | lt_pf |
| lt_pts | lt_pts |
| lt_reb | lt_reb |
| lt_stl | lt_stl |
| lt_td | lt_td |
| lt_tov | lt_tov |
| outcome | outcome |
| po_round | po_round |
| player_id | player_id |
| player_or_team | player_or_team |
| rookie_year | rookie_year |
| season | season - Min: '1983-84' |
| season_segment | season_segment |
| season_type | season_type |
| starter_bench | starter_bench |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_team_id | vs_team_id |
| years_experience | years_experience |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguegamefinder(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: LeagueGameFinderResults

**LeagueGameFinderResults**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Game Finder Functions: wnba_playergamestreakfinder(), wnba_teamgamestreakfinder()

---

# wnba_leaguegamelog Get WNBA Stats API League Game Log

---

## Description

**Get WNBA Stats API League Game Log**

**Get WNBA Stats API League Game Log**

## Usage

```
wnba_leaguegamelog(
  counter = 0,
  date_from = "",
  date_to = "",
  direction = "ASC",
  league_id = "10",
  player_or_team = "T",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  sorter = "DATE",
  ...
)
```

## Arguments

| | |
|---|---|
| counter | counter |
| date_from | date_from |
| date_to | date_to |
| direction | direction |
| league_id | league_id. Default '10' (WNBA). Use '00' for NBA or '20' for G-League data on the same endpoint. |
| player_or_team | player_or_team |
| season | season |
| season_type | season_type |
| sorter | sorter |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

Player/Team Boxscores

```
wnba_leaguegamelog(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: LeagueGameLog

**LeagueGameLog**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchup wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

`wnba_leagueleaders` **Get WNBA Stats API League Leaders**

## Description

**Get WNBA Stats API League Leaders**

**Get WNBA Stats API League Leaders**

## Usage

```
wnba_leagueleaders(
  active_flag = "",
  league_id = "10",
  per_mode = "Totals",
  scope = "S",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  stat_category = "PTS",
  ...
)
```

## Arguments

| | |
|---|---|
| `active_flag` | Active Flag |
| `league_id` | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| `per_mode` | Per Mode - Totals, PerGame, Per48 |
| `scope` | Scope - RS, S, Rookies |
| `season` | Season - format 2020-21 |
| `season_type` | Season Type - Regular Season, Playoffs |
| `stat_category` | Stat Category: PTS, REB, AST, FG_PCT, FT_PCT, FG3_PCT, STL, BLK |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leagueleaders(league_id = '10')
```

## Value

Returns a named list of data frames: LeagueLeaders

**LeagueLeaders**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Leaders Functions: wnba_alltimeleadersgrids(), wnba_assistleaders(), wnba_assisttracker(), wnba_homepagewidget()

---

wnba_leaguelineupviz **Get WNBA Stats API League Lineup Visual Data**

---

## Description

**Get WNBA Stats API League Lineup Visual Data**

**Get WNBA Stats API League Lineup Visual Data**

## Usage

```
wnba_leaguelineupviz(
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  game_segment = "",
  group_quantity = 5,
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  minutes_min = 10,
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| division | division |
| game_segment | game_segment |
| group_quantity | group_quantity |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| minutes_min | minutes_min |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leaguelineupviz(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: LeagueLineupViz

**LeagueLineupViz**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups( wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Lineup Functions: wnba_gamerotation(), wnba_leaguedashlineups(), wnba_possession_lineups(), wnba_rapm(), wnba_shot_zones(), wnba_shotchartlineupdetail(), wnba_teamdashlineups()

---

`wnba_leagueplayerondetails`

# Get WNBA Stats API League Player On/Off Details

---

## Description

**Get WNBA Stats API League Player On/Off Details**

**Get WNBA Stats API League Player On/Off Details**

## Usage

```
wnba_leagueplayerondetails(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  team_id = "1611661313",
  vs_conference = "",
  vs_division = "",
```

```
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_leagueplayerondetails(team_id = '1611661313', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: PlayersOnCourtLeaguePlayerDetails

**PlayersOnCourtLeaguePlayerDetails**

Columns as documented in the shared wnba_on_off_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycoll wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

wnba_leagueseasonmatchups

# Get WNBA Stats API League Season Matchups

---

## Description

**Get WNBA Stats API League Season Matchups**

**Get WNBA Stats API League Season Matchups**

## Usage

```
wnba_leagueseasonmatchups(
  def_player_id = "",
  def_team_id = "",
  league_id = "10",
  off_player_id = "",
  off_team_id = "",
  per_mode = "Totals",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| def_player_id | def_player_id |
| def_team_id | def_team_id |
| league_id | league_id |
| off_player_id | off_player_id |
| off_team_id | off_team_id |

| | |
|---|---|
| per_mode | per_mode |
| season | season |
| season_type | season_type |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

(No Matchups Data for WNBA yet, so defunct)

```
wnba_leagueseasonmatchups(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Returns a named list of data frames: SeasonMatchups

**SeasonMatchups**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leaguestandingsv3(), wnba_shotchartleaguewide()

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_playerawards(), wnba_playercareerbycollege(), wnba_playercareerbycol
wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet
wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_leaguestandingsv3`

**Get WNBA Stats API League Standings V3**

---

## Description

**Get WNBA Stats API League Standings V3**

**Get WNBA Stats API League Standings V3**

## Usage

```
wnba_leaguestandingsv3(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  season_year = "",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | league_id |
| season | season |
| season_type | season_type |
| season_year | season_year |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

League Standings

```
wnba_leaguestandingsv3(league_id = '10', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: Standings

**Standings**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_shotchartleaguewide()

`wnba_live_boxscore` **Get WNBA Stats API Live Boxscore**

## Description

**Get WNBA Stats API Live Boxscore**

**Get WNBA Stats API Live Boxscore**

## Usage

```
wnba_live_boxscore(game_id, ...)
```

## Arguments

| | |
|---|---|
| `game_id` | Game ID |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_live_boxscore(game_id = "1022200034")
```

## Value

Returns a named list of data frames: game_details, arena, officials, home_team_boxscore, away_team_boxscore, home_team_player_boxscore, away_team_player_boxscore, home_team_linescores, away_team_linescores

**game_details**

See the HTML help or pkgdown reference for the column table.

**arena**

| col_name | types | description |
|---|---|---|
| arena_id | integer | Unique identifier for arena. |
| arena_name | character | Arena name. |
| arena_city | character | Arena city. |
| arena_state | character | Arena state. |
| arena_country | character | Arena country. |
| arena_timezone | character | Arena timezone. |

**officials**

| col_name | types | description |
|---|---|---|
| person_id | integer | Unique player identifier (V3 endpoints). |
| name | character | Display name. |
| name_i | character | Initialed name (e.g. 'A. Wilson'). |
| first_name | character | Player's first name. |

| | | |
|---|---|---|
| family_name | character | Player's family / last name. |
| jersey_num | character | Jersey number worn by the player. |
| assignment | character | Assignment. |

**home_team_boxscore**

See the HTML help or pkgdown reference for the column table.

**away_team_boxscore**

Same columns as the **home_team_boxscore** table above.

**home_team_player_boxscore**

See the HTML help or pkgdown reference for the column table.

**away_team_player_boxscore**

Same columns as the **home_team_player_boxscore** table above.

**home_team_linescores**

| col_name | types | description |
|---|---|---|
| period | integer | Period of the game (1-4 quarters; 5+ for OT). |
| period_type | character | Period type. |
| score | integer | Final score. |

**away_team_linescores**

Same columns as the **home_team_linescores** table above.

## Author(s)

Saiem Gilani

---

`wnba_live_pbp` **Get WNBA Stats API Live play-by-play**

---

## Description

**Get WNBA Stats API Live play-by-play**

**Get WNBA Stats API Live play-by-play**

## Usage

```
wnba_live_pbp(game_id, ...)
```

## Arguments

| | |
|---|---|
| `game_id` | Game ID |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_live_pbp(game_id = "1022200034")
```

## Value

Returns a data frame: PlayByPlay

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA PBP Functions: `wnba_data_pbp()`, `wnba_pbp()`, `wnba_pbps()`, `wnba_playbyplayv3()`

Other WNBA Live Functions: `wnba_todays_scoreboard()`

---

`wnba_pbp` **Get WNBA Stats API play-by-play**

---

## Description

**Get WNBA Stats API play-by-play**

**Get WNBA Stats API play-by-play**

## Usage

```
wnba_pbp(game_id, on_court = TRUE, version = "v3", p = NULL, ...)
```

## Arguments

| | |
|---|---|
| `game_id` | Game ID |
| `on_court` | IF TRUE will be added ID of players on court |
| `version` | Play-by-play version - `"v3"` (default) or `"v2"`. V3 returns richer data with shot coordinates, shot values, and V3 action types. V2 is available from 2016-17 onwards. |
| `p` | Progress bar |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_pbp(game_id = "1022200034", on_court = TRUE)
```

## Value

Returns a data frame: PlayByPlay

See the HTML help or pkgdown reference for the column table.

## See Also

Other WNBA PBP Functions: `wnba_data_pbp()`, `wnba_live_pbp()`, `wnba_pbps()`, `wnba_playbyplayv3()`

---

`wnba_pbps` **Get WNBA Stats API play-by-play (Multiple Games)**

---

## Description

**Get WNBA Stats API play-by-play (Multiple Games)**

**Get WNBA Stats API play-by-play (Multiple Games)**

## Usage

```
wnba_pbps(
  game_ids = NULL,
  on_court = TRUE,
  version = "v3",
  nest_data = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `game_ids` | Game IDs |
| `on_court` | IF TRUE will be added ID of players on court |
| `version` | Play-by-play version - `"v3"` (default) or `"v2"`. V3 returns richer data with shot coordinates, shot values, and V3 action types. V2 is available from 2016-17 onwards. |
| `nest_data` | If TRUE returns nested data by game |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
y <- c("1022200034", "1022200035" )

wnba_pbps(game_ids = y, version = "v2")
```

## Value

Returns a data frame: PlayByPlay

Columns as documented in the shared wnba_players_on_court_schema table.

## Author(s)

Jason Lee

## See Also

Other WNBA PBP Functions: wnba_data_pbp(), wnba_live_pbp(), wnba_pbp(), wnba_playbyplayv3()

---

wnba_playbyplayv3 **Get WNBA Stats API Play-by-Play V3**

---

## Description

**Get WNBA Stats API Play-by-Play V3**

**Get WNBA Stats API Play-by-Play V3**

## Usage

```
wnba_playbyplayv3(game_id, start_period = 0, end_period = 0, ...)
```

## Arguments

| | |
|---|---|
| game_id | Game ID - 10-digit zero-padded ID (e.g., '1022200034') |
| start_period | Start period filter (default: 0 = all periods). |
| end_period | End period filter (default: 0 = all periods). |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playbyplayv3(game_id = "1022200034")
```

## Value

Returns a named list of data frames: PlayByPlay, AvailableVideo

**PlayByPlay**

See the HTML help or pkgdown reference for the column table.

**AvailableVideo**

| col_name | types | description |
|---|---|---|
| video_available | integer | Video available. |

## Author(s)

Saiem Gilani

## See Also

Other WNBA PBP Functions: wnba_data_pbp(), wnba_live_pbp(), wnba_pbp(), wnba_pbps()

## wnba_player_crosswalk Get the WNBA cross-source player crosswalk

### Description

Build a wide, one-row-per-player-per-team crosswalk linking ESPN, WNBA Stats, and Fox player identities for a season. Matching is deterministic: normalized exact name within a team block, then Jaro-Winkler fuzzy with jersey/DOB tiebreakers. Yahoo columns are NA placeholders.

### Usage

```
wnba_player_crosswalk(
  season = most_recent_wnba_season(),
  min_confidence = 0.92
)
```

### Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| min_confidence | Jaro-Winkler similarity floor for fuzzy matches (default 0.92). |

### Value

A wehoop_data tibble, one row per player per team (ESPN-anchored).

### See Also

Other WNBA Crosswalk Functions: load_wnba_team_crosswalk(), wnba_schedule_crosswalk(), wnba_team_crosswalk()

### Examples

```
try(wnba_player_crosswalk(season = 2024))
```

## wnba_playerawards Get WNBA Stats API Player Awards

### Description

**Get WNBA Stats API Player Awards**

**Get WNBA Stats API Player Awards**

### Usage

```
wnba_playerawards(player_id, ...)
```

### Arguments

| | |
|---|---|
| player_id | Player ID |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

Player Awards

```
wnba_playerawards(player_id = '1628932')
```

### Value

Return a named list of data frames: PlayerAwards

**PlayerAwards**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playercareerbycollege(), wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

## wnba_playercareerbycollege
## Get WNBA Stats API Player Career By College

---

### Description

**Get WNBA Stats API Player Career By College**

**Get WNBA Stats API Player Career By College**

## Usage

```
wnba_playercareerbycollege(
  college = "Florida State",
  league_id = "10",
  per_mode = "Totals",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| college | College Name |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| per_mode | Per Mode - PerGame, Totals |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playercareerbycollege(college = 'Florida State', per_mode = 'PerGame')
```

## Value

Return a named list of data frames: PlayerCareerByCollege

**PlayerCareerByCollege**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

`wnba_playercareerbycollegerollup`

## Get WNBA Stats API Player Career By College Rollup

### Description

**Get WNBA Stats API Player Career By College Rollup**

**Get WNBA Stats API Player Career By College Rollup**

### Usage

```
wnba_playercareerbycollegerollup(
  league_id = "10",
  per_mode = "Totals",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

### Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `per_mode` | Per Mode - PerGame, Totals |
| `season` | Season - format 2020-21 |
| `season_type` | Season Type - Regular Season, Playoffs, All-Star |
| `...` | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_playercareerbycollegerollup(per_mode = 'Totals')
```

### Value

Return a named list of data frames: East, Midwest, South, West

**East**

See the HTML help or pkgdown reference for the column table.

**South**

Same columns as the **East** table above.

**Midwest**

Same columns as the **East** table above.

**West**

Same columns as the **East** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmet wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_playercareerstats`

# Get WNBA Stats API Player Career Stats

---

## Description

**Get WNBA Stats API Player Career Stats**

**Get WNBA Stats API Player Career Stats**

## Usage

```
wnba_playercareerstats(
  league_id = "10",
  per_mode = "Totals",
  player_id = "1628932",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `per_mode` | Per Mode - PerGame, Totals |
| `player_id` | Player ID |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playercareerstats(player_id = '1628932')
```

## Value

Return a named list of data frames: CareerTotalsAllStarSeason, CareerTotalsCollegeSeason, CareerTotalsPostSeason, CareerTotalsRegularSeason, SeasonRankingsPostSeason, SeasonRankingsRegularSeason, SeasonTotalsAllStarSeason, SeasonTotalsCollegeSeason, SeasonTotalsPostSeason, SeasonTotalsRegularSeason

**SeasonTotalsRegularSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsRegularSeason**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| LEAGUE_ID | character | League identifier ('10' = WNBA). |
| Team_ID | character | Unique team identifier. |
| GP | character | Games played. |
| GS | character | Games started. |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| TOV | character | Turnovers. |
| PF | character | Personal fouls. |
| PTS | character | Points scored. |

**SeasonTotalsPostSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsPostSeason**

Same columns as the **CareerTotalsRegularSeason** table above.

**SeasonTotalsAllStarSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsAllStarSeason**

Same columns as the **CareerTotalsRegularSeason** table above.

**SeasonTotalsCollegeSeason**

**CareerTotalsCollegeSeason**

**SeasonTotalsShowcaseSeason**

**CareerTotalsShowcaseSeason**

**SeasonRankingsRegularSeason**

Columns as documented in the shared wnba_playercareerstats_season_rankings_schema table.

**SeasonRankingsPostSeason**

Columns as documented in the shared wnba_playercareerstats_season_rankings_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

# wnba_playercompare Get WNBA Stats API Player Compare

---

## Description

**Get WNBA Stats API Player Compare**

**Get WNBA Stats API Player Compare**

## Usage

```
wnba_playercompare(
  conference = "",
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
```

```
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id_list = "100720,202250,204319,1627668,1628931",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  vs_player_id_list = "202252,203399,1631022,1628878,204333",
  ...
)
```

## Arguments

| | |
|---|---|
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| player_id_list | Player ID |
| plus_minus | plus_minus |
| rank | rank |
| season | Season - format 2020-21 |
| season_segment | season_segment |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| shot_clock_range | shot_clock_range |
| vs_conference | vs_conference |

vs_division vs_division

vs_player_id_list
: vs_player_id_list

... : Additional arguments passed to an underlying function like httr.

## Details

```
wnba_playercompare(player_id_list = '100720,202250,204319,1627668,1628931', vs_player_id_list = '202
```

## Value

Return a named list of data frames: Individual, OverallCompare

**OverallCompare**

See the HTML help or pkgdown reference for the column table.

**Individual**

Same columns as the **OverallCompare** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

# wnba_playerdashboardbyclutch

**Get WNBA Stats API Player Dashboard by Clutch Splits**

---

## Description

**Get WNBA Stats API Player Dashboard by Clutch Splits**

**Get WNBA Stats API Player Dashboard by Clutch Splits**

**Get WNBA Stats API Player Dashboard by Game Splits**

**Get WNBA Stats API Player Dashboard by General Splits**

**Get WNBA Stats API Player Dashboard by Last N Games**

**Get WNBA Stats API Player Dashboard by Opponent**

**Get WNBA Stats API Player Dashboard by Shooting Splits**

**Get WNBA Stats API Player Dashboard by Team Performance**

**Get WNBA Stats API Player Dashboard Year over Year**

## Usage

```
wnba_playerdashboardbyclutch(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbygamesplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
```

```
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbygeneralsplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbylastngames(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
```

```
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbyopponent(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbyshootingsplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
```

```
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbyteamperformance(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_playerdashboardbyyearoveryear(
  date_from = "",
```

```
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| player_id | player_id |
| plus_minus | plus_minus |

| | |
|---|---|
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerdashboardbyclutch(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbygamesplits(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbygeneralsplits(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbylastngames(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbyopponent(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbyshootingsplits(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbyteamperformance(player_id = '1628932', season = most_recent_wnba_season() - 1)

wnba_playerdashboardbyyearoveryear(player_id = '1628932', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: Last10Sec3Point2PlayerDashboard, Last10Sec3PointPlayerDashboard, Last1Min5PointPlayerDashboard, Last1MinPlusMinus5PointPlayerDashboard, Last30Sec3Point2PlayerDashboard, Last30Sec3PointPlayerDashboard, Last3Min5PointPlayerDashboard, Last3MinPlusMinus5PointPlayerDashboard, Last5Min5PointPlayerDashboard, Last5MinPlusMinus5PointPlayerDashboard, OverallPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last5Min5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last3Min5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last1Min5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last30Sec3PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last10Sec3PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last5MinPlusMinus5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last3MinPlusMinus5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last1MinPlusMinus5PointPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last30Sec3Point2PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last10Sec3Point2PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

Return a named list of data frames: ByActualMarginPlayerDashboard, ByHalfPlayerDashboard, ByPeriodPlayerDashboard, ByScoreMarginPlayerDashboard, OverallPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ByHalfPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ByPeriodPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ByScoreMarginPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ByActualMarginPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

Return a named list of data frames: DaysRestPlayerDashboard, LocationPlayerDashboard, MonthPlayerDashboard, OverallPlayerDashboard, PrePostAllStarPlayerDashboard, StartingPosition, WinsLossesPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**LocationPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**WinsLossesPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**MonthPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**PrePostAllStarPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**StartingPosition**

Columns as documented in the shared wnba_player_dashboard_schema table.

**DaysRestPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

Return a named list of data frames: GameNumberPlayerDashboard, Last10PlayerDashboard, Last15PlayerDashboard, Last20PlayerDashboard, Last5PlayerDashboard, OverallPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last5PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last10PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last15PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**Last20PlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**GameNumberPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

Return a named list of data frames: ConferencePlayerDashboard, DivisionPlayerDashboard, OpponentPlayerDashboard, OverallPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ConferencePlayerDashboard**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| GP | character | Games played. |
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |

| | | |
|---|---|---|
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| NBA_FANTASY_PTS | character | |
| DD2 | character | |
| TD3 | character | |
| WNBA_FANTASY_PTS | character | |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |
| FGM_RANK | character | |
| FGA_RANK | character | |
| FG_PCT_RANK | character | |
| FG3M_RANK | character | |
| FG3A_RANK | character | |
| FG3_PCT_RANK | character | |
| FTM_RANK | character | |
| FTA_RANK | character | |
| FT_PCT_RANK | character | |
| OREB_RANK | character | |
| DREB_RANK | character | |
| REB_RANK | character | |
| AST_RANK | character | |
| TOV_RANK | character | |
| STL_RANK | character | |
| BLK_RANK | character | |
| BLKA_RANK | character | |
| PF_RANK | character | |
| PFD_RANK | character | |
| PTS_RANK | character | |
| PLUS_MINUS_RANK | character | |
| WBA_FANTASY_PTS_RANK | character | |
| DD2_RANK | character | |
| TD3_RANK | character | |
| WNBA_FANTASY_PTS_RANK | character | |

**DivisionPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**OpponentPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

Return a named list of data frames: AssistedBy, AssitedShotPlayerDashboard, OverallPlayerDashboard, Shot5FTPlayerDashboard, Shot8FTPlayerDashboard, ShotAreaPlayerDashboard, ShotTypePlayerDashboard, ShotTypeSummaryPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**Shot5FTPlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**Shot8FTPlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**ShotAreaPlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**AssitedShotPlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**ShotTypeSummaryPlayerDashboard**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| EFG_PCT | character | Effective field goal percentage (0-1). |
| BLKA | character | |
| PCT_AST_2PM | character | |
| PCT_UAST_2PM | character | |
| PCT_AST_3PM | character | |
| PCT_UAST_3PM | character | |
| PCT_AST_FGM | character | |
| PCT_UAST_FGM | character | |

**ShotTypePlayerDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**AssistedBy**

Columns as documented in the shared wnba_shooting_splits_assisted_by_schema table.

Return a named list of data frames: OverallPlayerDashboard, PointsScoredPlayerDashboard, PointsAgainstPlayerDashboard, ScoreDifferentialPlayerDashboard

**OverallPlayerDashboard**

Columns as documented in the shared wnba_player_dashboard_schema table.

**ScoreDifferentialPlayerDashboard**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE_ORDER | character | |
| GROUP_VALUE | character | |
| GROUP_VALUE_2 | character | |
| GP | character | Games played. |
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| NBA_FANTASY_PTS | character | |
| DD2 | character | |
| TD3 | character | |
| WNBA_FANTASY_PTS | character | |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |

| | |
|---|---|
| FGM_RANK | character |
| FGA_RANK | character |
| FG_PCT_RANK | character |
| FG3M_RANK | character |
| FG3A_RANK | character |
| FG3_PCT_RANK | character |
| FTM_RANK | character |
| FTA_RANK | character |
| FT_PCT_RANK | character |
| OREB_RANK | character |
| DREB_RANK | character |
| REB_RANK | character |
| AST_RANK | character |
| TOV_RANK | character |
| STL_RANK | character |
| BLK_RANK | character |
| BLKA_RANK | character |
| PF_RANK | character |
| PFD_RANK | character |
| PTS_RANK | character |
| PLUS_MINUS_RANK | character |
| NBA_FANTASY_PTS_RANK | character |
| DD2_RANK | character |
| TD3_RANK | character |
| WNBA_FANTASY_PTS_RANK | character |

**PointsScoredPlayerDashboard**

Same columns as the **ScoreDifferentialPlayerDashboard** table above.

**PontsAgainstPlayerDashboard**

Same columns as the **ScoreDifferentialPlayerDashboard** table above.

Return a named list of data frames: ByYearPlayerDashboard, OverallPlayerDashboard

**OverallPlayerDashboard**

See the HTML help or pkgdown reference for the column table.

**ByYearPlayerDashboard**

Same columns as the **OverallPlayerDashboard** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: `wnba_infographicfanduelplayer()`, `wnba_leaguedashplayerbiostats()`, `wnba_leaguedashplayerclutch()`, `wnba_leaguedashplayershotlocations()`, `wnba_leaguedashplayerstats()`, `wnba_leagueplayerondetails()`, `wnba_leagueseasonmatchups()`, `wnba_playerawards()`, `wnba_playercareerbycoll`

wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Clutch Functions: wnba_leaguedashteamclutch(), wnba_teamdashboardbyclutch()

Other WNBA Shooting Functions: wnba_leaguedashplayershotlocations(), wnba_leaguedashteamshotlocations(), wnba_shotchartdetail(), wnba_shotchartleaguewide(), wnba_shotchartlineupdetail(), wnba_teamdashboardbyclutch()

---

`wnba_playerestimatedmetrics`

## Get WNBA Stats API Player Estimated Metrics

---

### Description

**Get WNBA Stats API Player Estimated Metrics**

**Get WNBA Stats API Player Estimated Metrics**

### Usage

```
wnba_playerestimatedmetrics(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

### Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `season` | Season - format 2020-21 |
| `season_type` | Season Type - Regular Season, Playoffs, All-Star |
| `...` | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_playerestimatedmetrics()
```

### Value

Return a named list of data frames: PlayerEstimatedMetrics

**PlayerEstimatedMetrics**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_playerfantasyprofile`

## Get WNBA Stats API Player Fantasy Profile

---

### Description

**Get WNBA Stats API Player Fantasy Profile**

**Get WNBA Stats API Player Fantasy Profile**

### Usage

```
wnba_playerfantasyprofile(
  league_id = "10",
  measure_type = "Base",
  pace_adjust = "N",
  per_mode = "Totals",
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

### Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `measure_type` | measure_type |
| `pace_adjust` | Pace Adjustment - Y/N |
| `per_mode` | Per Mode - PerGame, Totals |
| `player_id` | Player ID |
| `plus_minus` | Plus Minus - Y/N |

| | |
|---|---|
| rank | Rank - Y/N |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerfantasyprofile(player_id = '1628932')
```

## Value

Return a named list of data frames: DaysRestModified, LastNGames, Location, Opponent, Overall

**Overall**

See the HTML help or pkgdown reference for the column table.

**Location**

Same columns as the **Overall** table above.

**LastNGames**

Same columns as the **Overall** table above.

**DaysRestModified**

See the HTML help or pkgdown reference for the column table.

**Opponent**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Fantasy Functions: wnba_fantasywidget(), wnba_infographicfanduelplayer(), wnba_playerfantasyprofilebargraph()

wnba_playerfantasyprofilebargraph

**Get WNBA Stats API Player Fantasy Profile Bar Graph**

## Description

**Get WNBA Stats API Player Fantasy Profile Bar Graph**

**Get WNBA Stats API Player Fantasy Profile Bar Graph**

## Usage

```
wnba_playerfantasyprofilebargraph(
  league_id = "10",
  player_id = "1628932",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| player_id | Player ID |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerfantasyprofilebargraph(player_id = '1628932')
```

## Value

Return a named list of data frames: LastFiveGamesAvg, SeasonAvg

**SeasonAvg**

See the HTML help or pkgdown reference for the column table.

**LastFiveGamesAvg**

Same columns as the **SeasonAvg** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Fantasy Functions: wnba_fantasywidget(), wnba_infographicfanduelplayer(), wnba_playerfantasyprofile()

---

# wnba_playergamelog Get WNBA Stats API Player Game Log

---

## Description

**Get WNBA Stats API Player Game Log**

**Get WNBA Stats API Player Game Log**

## Usage

```
wnba_playergamelog(
  date_from = "",
  date_to = "",
  league_id = "10",
  player_id = "1628932",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| player_id | Player ID |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playergamelog(player_id = '1628932')
```

## Value

Return a named list of data frames: PlayerGameLog

**PlayerGameLog**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

# wnba_playergamelogs Get WNBA Stats API Player Game Logs

---

## Description

**Get WNBA Stats API Player Game Logs**

**Get WNBA Stats API Player Game Logs**

## Usage

```
wnba_playergamelogs(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  season = most_recent_wnba_season() - 1,
```

```
  season_segment = "",
  season_type = "Regular Season",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_id | Player ID |
| season | Season - format 2020-21 |
| season_segment | season_segment |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playergamelogs(player_id = '1628932')
```

## Value

Return a named list of data frames: PlayerGameLogs

**PlayerGameLogs**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycol wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_playergamestreakfinder`

# Get WNBA Stats API Player Game Streak Finder

---

## Description

**Get WNBA Stats API Player Game Streak Finder**

**Get WNBA Stats API Player Game Streak Finder**

## Usage

```
wnba_playergamestreakfinder(
  active_streaks_only = "",
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
  draft_year = "",
  draft_team_id = "",
  draft_round = "",
  draft_number = "",
  et_ast = "",
  et_blk = "",
  et_dd = "",
  et_dreb = "",
  et_fg3a = "",
  et_fg3m = "",
  et_fg3_pct = "",
  et_fga = "",
  et_fgm = "",
  et_fg_pct = "",
  et_fta = "",
```

```
    et_ftm = "",
    et_ft_pct = "",
    et_minutes = "",
    et_oreb = "",
    et_pf = "",
    et_pts = "",
    et_reb = "",
    et_stl = "",
    et_td = "",
    et_tov = "",
    game_id = "",
    gt_ast = "",
    gt_blk = "",
    gt_dd = "",
    gt_dreb = "",
    gt_fg3a = "",
    gt_fg3m = "",
    gt_fg3_pct = "",
    gt_fga = "",
    gt_fgm = "",
    gt_fg_pct = "",
    gt_fta = "",
    gt_ftm = "",
    gt_ft_pct = "",
    gt_minutes = "",
    gt_oreb = "",
    gt_pf = "",
    gt_pts = "",
    gt_reb = "",
    gt_stl = "",
    gt_td = "",
    gt_tov = "",
    league_id = "10",
    location = "",
    lt_ast = "",
    lt_blk = "",
    lt_dd = "",
    lt_dreb = "",
    lt_fg3a = "",
    lt_fg3m = "",
    lt_fg3_pct = "",
    lt_fga = "",
    lt_fgm = "",
    lt_fg_pct = "",
    lt_fta = "",
    lt_ftm = "",
    lt_ft_pct = "",
    lt_minutes = "",
```

```
  lt_oreb = "",
  lt_pf = "",
  lt_pts = "",
  lt_reb = "",
  lt_stl = "",
  lt_td = "",
  lt_tov = "",
  min_games = "",
  outcome = "",
  po_round = "",
  player_id = "",
  rookie_year = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  starter_bench = "",
  team_id = "",
  vs_conference = "",
  vs_division = "",
  vs_team_id = "",
  years_experience = "",
  ...
)
```

## Arguments

| | |
|---|---|
| active_streaks_only | active_streaks_only |
| conference | conference |
| date_from | date_from |
| date_to | date_to |
| division | division |
| draft_year | draft_year |
| draft_team_id | draft_team_id |
| draft_round | draft_round |
| draft_number | draft_number |
| et_ast | et_ast |
| et_blk | et_blk |
| et_dd | et_dd |
| et_dreb | et_dreb |
| et_fg3a | et_fg3a |
| et_fg3m | et_fg3m |
| et_fg3_pct | et_fg3_pct |
| et_fga | et_fga |

| | |
|---|---|
| et_fgm | et_fgm |
| et_fg_pct | et_fg_pct |
| et_fta | et_fta |
| et_ftm | et_ftm |
| et_ft_pct | et_ft_pct |
| et_minutes | et_minutes |
| et_oreb | et_oreb |
| et_pf | et_pf |
| et_pts | et_pts |
| et_reb | et_reb |
| et_stl | et_stl |
| et_td | et_td |
| et_tov | et_tov |
| game_id | game_id |
| gt_ast | gt_ast |
| gt_blk | gt_blk |
| gt_dd | gt_dd |
| gt_dreb | gt_dreb |
| gt_fg3a | gt_fg3a |
| gt_fg3m | gt_fg3m |
| gt_fg3_pct | gt_fg3_pct |
| gt_fga | gt_fga |
| gt_fgm | gt_fgm |
| gt_fg_pct | gt_fg_pct |
| gt_fta | gt_fta |
| gt_ftm | gt_ftm |
| gt_ft_pct | gt_ft_pct |
| gt_minutes | gt_minutes |
| gt_oreb | gt_oreb |
| gt_pf | gt_pf |
| gt_pts | gt_pts |
| gt_reb | gt_reb |
| gt_stl | gt_stl |
| gt_td | gt_td |
| gt_tov | gt_tov |
| league_id | league_id |
| location | location |

| | |
|---|---|
| lt_ast | lt_ast |
| lt_blk | lt_blk |
| lt_dd | lt_dd |
| lt_dreb | lt_dreb |
| lt_fg3a | lt_fg3a |
| lt_fg3m | lt_fg3m |
| lt_fg3_pct | lt_fg3_pct |
| lt_fga | lt_fga |
| lt_fgm | lt_fgm |
| lt_fg_pct | lt_fg_pct |
| lt_fta | lt_fta |
| lt_ftm | lt_ftm |
| lt_ft_pct | lt_ft_pct |
| lt_minutes | lt_minutes |
| lt_oreb | lt_oreb |
| lt_pf | lt_pf |
| lt_pts | lt_pts |
| lt_reb | lt_reb |
| lt_stl | lt_stl |
| lt_td | lt_td |
| lt_tov | lt_tov |
| min_games | min_games |
| outcome | outcome |
| po_round | po_round |
| player_id | player_id |
| rookie_year | rookie_year |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| starter_bench | starter_bench |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_team_id | vs_team_id |
| years_experience | years_experience |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playergamestreakfinder()
```

## Value

Return a named list of data frames: PlayerGameStreakFinderResults

**PlayerGameStreakFinderResults**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

Other WNBA Game Finder Functions: wnba_leaguegamefinder(), wnba_teamgamestreakfinder()

---

# wnba_playerheadshot Get WNBA Stats API Player Head-shot

---

## Description

**Get WNBA Stats API Player Head-shot**

**Get WNBA Stats API Player Head-shot**

## Usage

```
wnba_playerheadshot(player_id = "1628932", ...)
```

## Arguments

| | |
|---|---|
| player_id | Player ID |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerheadshot(player_id = '1628932')
```

## Value

Returns a url of the png for the player_id selected

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerindex(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_playerindex` **Get WNBA Stats API Player Index**

---

## Description

**Get WNBA Stats API Player Index**

**Get WNBA Stats API Player Index**

## Usage

```
wnba_playerindex(
  college = "",
  country = "",
  draft_pick = "",
  draft_round = "",
  draft_year = "",
  height = "",
  historical = 1,
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  team_id = "0",
  weight = "",
  ...
)
```

## Arguments

| | |
|---|---|
| college | Player College |
| country | Player Country |
| draft_pick | Draft Pick |
| draft_round | Draft Round |
| draft_year | Draft Year |
| height | Player Height |
| historical | Whether to include only current players (0) or all historical (1). |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | Team ID. Default: 0 (all teams). |
| weight | Player weight |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerindex()
```

## Value

Return a named list of data frames: PlayerIndex

**PlayerIndex**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playernextngames(), wnba_playerprofilev2(), wnba_playervsplayer()

wnba_playernextngames **Get WNBA Stats API Player Next N Games**

## Description

**Get WNBA Stats API Player Next N Games**

**Get WNBA Stats API Player Next N Games**

## Usage

```
wnba_playernextngames(
  league_id = "10",
  number_of_games = 2147483647,
  player_id = "1628932",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| number_of_games | N in number of games |
| player_id | Player ID |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playernextngames(player_id = '1628932', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: NextNGames

**NextNGames**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playerprofilev2(), wnba_playervsplayer()

---

`wnba_playerprofilev2` **Get WNBA Stats API Player Profile V2**

---

## Description

**Get WNBA Stats API Player Profile V2**

**Get WNBA Stats API Player Profile V2**

## Usage

```
wnba_playerprofilev2(
  league_id = "10",
  per_mode = "Totals",
  player_id = "1628932",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| `per_mode` | Season - format 2020-21 |
| `player_id` | Player ID |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playerprofilev2(player_id = '1628932')
```

## Value

Return a named list of data frames: CareerHighs, CareerTotalsAllStarSeason, CareerTotalsCollegeSeason, CareerTotalsPostSeason, CareerTotalsPreseason, CareerTotalsRegularSeason, NextGame, SeasonHighs, SeasonRankingsPostSeason, SeasonRankingsRegularSeason, SeasonTotalsAllStarSeason, SeasonTotalsCollegeSeason, SeasonTotalsPostSeason, SeasonTotalsPreseason, SeasonTotalsRegularSeason

**SeasonTotalsRegularSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsRegularSeason**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| LEAGUE_ID | character | League identifier ('10' = WNBA). |
| TEAM_ID | character | Unique team identifier. |
| GP | character | Games played. |
| GS | character | Games started. |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| TOV | character | Turnovers. |
| PF | character | Personal fouls. |
| PTS | character | Points scored. |

**SeasonTotalsPostSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsPostSeason**

Same columns as the **CareerTotalsRegularSeason** table above.

**SeasonTotalsAllStarSeason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsAllStarSeason**

Same columns as the **CareerTotalsRegularSeason** table above.

**SeasonTotalsCollegeSeason**

**CareerTotalsCollegeSeason**

**SeasonTotalsPreseason**

Columns as documented in the shared wnba_playercareerstats_season_totals_schema table.

**CareerTotalsPreseason**

Same columns as the **CareerTotalsRegularSeason** table above.

**SeasonRankingsRegularSeason**

Columns as documented in the shared wnba_playercareerstats_season_rankings_schema table.

**SeasonRankingsPostSeason**

Columns as documented in the shared wnba_playercareerstats_season_rankings_schema table.

**SeasonHighs**

| col_name | types | description |
|---|---|---|
| PLAYER_ID | character | Unique player identifier. |
| GAME_ID | character | Unique game identifier. |
| GAME_DATE | character | Game date (YYYY-MM-DD). |
| VS_TEAM_ID | character | |
| VS_TEAM_CITY | character | |
| VS_TEAM_NAME | character | |
| VS_TEAM_ABBREVIATION | character | |
| STAT | character | |
| STAT_VALUE | character | Stat value. |
| STAT_ORDER | character | |
| DATE_EST | character | |

**CareerHighs**

Same columns as the **SeasonHighs** table above.

**NextGame**

| col_name | types | description |
|---|---|---|
| GAME_ID | character | Unique game identifier. |
| GAME_DATE | character | Game date (YYYY-MM-DD). |
| GAME_TIME | character | Game start time. |
| LOCATION | character | Filter results by game location. |
| PLAYER_TEAM_ID | character | |
| PLAYER_TEAM_CITY | character | |
| PLAYER_TEAM_NICKNAME | character | |
| PLAYER_TEAM_ABBREVIATION | character | |
| VS_TEAM_ID | character | |
| VS_TEAM_CITY | character | |
| VS_TEAM_NICKNAME | character | |
| VS_TEAM_ABBREVIATION | character | |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: wnba_infographicfanduelplayer(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_playerawards(), wnba_playercareerbycoll wnba_playercareerbycollegerollup(), wnba_playercareerstats(), wnba_playercompare(), wnba_playerdashboardbyclutch(), wnba_playerestimatedmetrics(), wnba_playerfantasyprofile(), wnba_playerfantasyprofilebargraph(), wnba_playergamelog(), wnba_playergamelogs(), wnba_playergamestreakfinder(), wnba_playerheadshot(), wnba_playerindex(), wnba_playernextngames(), wnba_playervsplayer()

---

`wnba_playervsplayer` **Get WNBA Stats API Player vs Player**

---

## Description

**Get WNBA Stats API Player vs Player**

**Get WNBA Stats API Player vs Player**

## Usage

```
wnba_playervsplayer(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "1628932",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  vs_conference = "",
  vs_division = "",
  vs_player_id = "1629488",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| player_id | Player ID |
| plus_minus | plus_minus |
| rank | rank |
| season | Season - format 2020-21 |
| season_segment | season_segment |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_player_id | vs_player_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_playervsplayer(player_id = '1628932', vs_player_id = '1629488')
```

## Value

Return a named list of data frames: OnOffCourt, Overall, PlayerInfo, ShotAreaOffCourt, ShotAreaOnCourt, ShotAreaOverall, ShotDistanceOffCourt, ShotDistanceOnCourt, ShotDistanceOverall, VsPlayerInfo

**Overall**

See the HTML help or pkgdown reference for the column table.

**OnOffCourt**

See the HTML help or pkgdown reference for the column table.

**ShotDistanceOverall**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |

**ShotDistanceOnCourt**

See the HTML help or pkgdown reference for the column table.

**ShotDistanceOffCourt**

Same columns as the **ShotDistanceOnCourt** table above.

**ShotAreaOverall**

Same columns as the **ShotDistanceOverall** table above.

**ShotAreaOnCourt**

Same columns as the **ShotDistanceOnCourt** table above.

**ShotAreaOffCourt**

Same columns as the **ShotDistanceOnCourt** table above.

**PlayerInfo**

See the HTML help or pkgdown reference for the column table.

**VsPlayerInfo**

Same columns as the **PlayerInfo** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Player Functions: `wnba_infographicfanduelplayer()`, `wnba_leaguedashplayerbiostats()`, `wnba_leaguedashplayerclutch()`, `wnba_leaguedashplayershotlocations()`, `wnba_leaguedashplayerstats()`, `wnba_leagueplayerondetails()`, `wnba_leagueseasonmatchups()`, `wnba_playerawards()`, `wnba_playercareerbycoll`
`wnba_playercareerbycollegerollup()`, `wnba_playercareerstats()`, `wnba_playercompare()`, `wnba_playerdashboardbyclutch()`, `wnba_playerestimatedmetrics()`, `wnba_playerfantasyprofile()`, `wnba_playerfantasyprofilebargraph()`, `wnba_playergamelog()`, `wnba_playergamelogs()`, `wnba_playergamestreakfinder()`, `wnba_playerheadshot()`, `wnba_playerindex()`, `wnba_playernextngames()`, `wnba_playerprofilev2()`

wnba_possession_lineups

## Get WNBA Possession-Level Stint Matrix

### Description

**Get WNBA Possession-Level Stint Matrix**

**Get WNBA Possession-Level Stint Matrix**

### Usage

```
wnba_possession_lineups(game_id, ...)
```

### Arguments

| | |
|---|---|
| game_id | WNBA Stats game id. 10-character zero-padded string (e.g., "1022400001"). |
| ... | reserved for future use. |

### Value

Returns a data frame with one row per possession:

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Lineup Functions: wnba_gamerotation(), wnba_leaguedashlineups(), wnba_leaguelineupviz(), wnba_rapm(), wnba_shot_zones(), wnba_shotchartlineupdetail(), wnba_teamdashlineups()

wnba_rapm

## Fit a Ridge-Regression RAPM Model from WNBA Possession Data

### Description

**Fit a Ridge-Regression RAPM Model from WNBA Possession Data**

**Fit a Ridge-Regression RAPM Model from WNBA Possession Data**

### Usage

```
wnba_rapm(possessions, ...)
```

### Arguments

| | |
|---|---|
| possessions | A possession-level stint matrix as produced by wnba_possession_lineups(), with columns off_player_1 through off_player_5, def_player_1 through def_player_5 (integer WNBA Stats person IDs), and points (numeric, points scored on that possession). |
| ... | Reserved for future keyword arguments (currently ignored). |

### Value

Returns a data.frame with one row per player:

| col_name | types | description |
|---|---|---|
| player_id | integer | WNBA Stats person ID. Rows are sorted ascending by player_id. |
| o_rapm | numeric | Offensive RAPM (per-100-possession points added on offense). Positive = better offensive player. |
| d_rapm | numeric | Defensive RAPM (per-100-possession points saved on defense). Positive = better defensive player (sig |
| rapm | numeric | Total RAPM = o_rapm + d_rapm. Positive = net positive impact. |
| off_poss | integer | Number of possessions the player appeared on offense. |
| def_poss | integer | Number of possessions the player appeared on defense. |

Returns a 0-row frame with the same schema when input is empty or all possessions have NA lineup cells (never-raise).

**Note:** RAPM is expressed in per-100-possession units. A **full season** of possessions (~3,000–5,000) is needed for statistically meaningful estimates. Results from a single game (~120–200 possessions) are highly unstable and are provided here for pipeline illustration only.

Results are **deterministic**: the cross-validation uses fixed, construction-based folds (not random), so repeated calls on the same possessions return identical output with no need to set a seed.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Lineup Functions: wnba_gamerotation(), wnba_leaguedashlineups(), wnba_leaguelineupviz(), wnba_possession_lineups(), wnba_shot_zones(), wnba_shotchartlineupdetail(), wnba_teamdashlineups()

---

## wnba_referee_assignments

**WNBA referee assignments for a given date**

---

### Description

Retrieves the referee crew assignments and replay-center officials for every WNBA game on date, from official.nba.com's internal get-game-officials endpoint (the same feed used by the NBA officiating pages). That endpoint returns NBA and G-League blocks in the same payload; this function extracts only "wnba". Port of the scraping logic in atlhawksfanatic/L2M (MIT, (c) 2019 atlhawksfanatic).

## Usage

```
wnba_referee_assignments(date, proxy = NULL)
```

## Arguments

| | |
|---|---|
| date | Date to fetch: a single Date, POSIXct/POSIXlt, or "YYYY-MM-DD" string. Datetimes are formatted in their own time zone, so a late-evening POSIXct keeps its calendar day. |
| proxy | Optional proxy config. NULL (default) falls back to getOption("wehoop.proxy"), then lets libcurl honor the standard http_proxy / https_proxy / no_proxy environment variables. A single URL string (e.g. "http://host:port") is forwarded to httr2::req_proxy(url = proxy). A named list is spread as keyword args into httr2::req_proxy() (url, port, username, password, auth). |

## Details

crew_position (1-4) is the feed's slot order; slot 1 is inferred to be the crew chief – official.nba.com does not label crew roles in this feed. The feed encodes the season as <type digit><START year> (e.g. "22026" is the 2026 regular season). season is that START year: a WNBA season sits inside one calendar year, so no end-year conversion is applied. season_type decodes the leading digit.

Failures are signalled as classed conditions, both inheriting wehoop_error:

- wehoop_no_data – HTTP 404, or HTTP 403 with an S3 <Code>AccessDenied</Code> XML body: official.nba.com has no report for date.
- wehoop_fetch_error – any other non-200 status (e.g. an Akamai HTML 403 block, or a rate limit / 5xx that outlived the retries), an empty or non-JSON body, JSON without the wnba Table/Table1 rows lists or with a Table row that has no game_id or a Table1 row that has no replaycenter_official name, or a transport failure (DNS, TLS, dropped connection). The feed carries the wnba block on every date, with zero rows on a day without games, so a missing block is never an empty day.

An invalid argument (date or proxy), including an impossible date such as "2026-02-31", is an ordinary error raised before any request.

Statuses 408, 429, 500, 502, 503 and 504 and transport failures are retried (3 attempts in total); 403 and 404 are definitive and never retried.

## Value

A named list of two wehoop_data tibbles.

officials – long format, one row per game x filled crew slot (WNBA crews have three officials, so slot 4 is usually absent):

See the HTML help or pkgdown reference for the column table.

replay_center – one row per replay-center official per game date:

See the HTML help or pkgdown reference for the column table.

A date with no WNBA games returns zero-row tibbles carrying this schema, not an error. A field the feed omits comes back as NA; it never drops the row or the column.

### Author(s)

Saiem Gilani

### Examples

```
try(wnba_referee_assignments(date = "2026-06-13"))
```

---

wnba_schedule **Get WNBA Stats API Season Schedule**

---

### Description

**Get WNBA Stats API Season Schedule**

**Get WNBA Stats API Season Schedule**

### Usage

```
wnba_schedule(league_id = "10", season = most_recent_wnba_season() - 1, ...)
```

### Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| season | Season |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_schedule(league_id = '10', season = most_recent_wnba_season() - 1)
```

### Value

Returns a tibble with the following columns:

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Schedule Functions: wnba_scoreboardv2(), wnba_scoreboardv3(), wnba_todays_scoreboard()

wnba_schedule_crosswalk

## Get the WNBA cross-source schedule crosswalk

### Description

Build a wide, one-row-per-game crosswalk linking ESPN and WNBA Stats game ids (with NA Fox/Yahoo placeholders) for a season. Dates from both sources are reduced to the local Eastern-Time game date before joining. Note: the WNBA Stats CDN serves the current season only, so the live builder is effectively current-season; historical coverage comes from cached release artifacts.

### Usage

```
wnba_schedule_crosswalk(season = most_recent_wnba_season())
```

### Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |

### Value

A wehoop_data tibble, one row per game (columns: season, season_type, game_date, resolved home_espn_team_id/away_espn_team_id, espn_game_id, wnba_game_id, wnba_game_code, Fox/Yahoo placeholders, match_method, match_confidence).

### See Also

Other WNBA Crosswalk Functions: load_wnba_team_crosswalk(), wnba_player_crosswalk(), wnba_team_crosswalk()

### Examples

```
try(wnba_schedule_crosswalk(season = 2024))
```

wnba_scoreboardv2 **Get WNBA Stats API Scoreboard V2**

### Description

**Get WNBA Stats API Scoreboard V2**

**Get WNBA Stats API Scoreboard V2**

## Usage

```
wnba_scoreboardv2(
  league_id = "10",
  game_date = "2022-07-20",
  day_offset = 0,
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WWNBA, '20': G-League |
| game_date | Game Date, format: 2022/05/17 |
| day_offset | Day Offset (integer 0,-1) |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_scoreboardv2(league_id = '10', game_date = '2022-07-20')
```

## Value

Return a named list of data frames: Available, EastConfStandingsByDay, GameHeader, LastMeeting, LineScore, SeriesStandings, TeamLeaders, TicketLinks, WestConfStandingsByDay, WinProbability

**GameHeader**

| col_name | types | description |
|---|---|---|
| GAME_DATE_EST | character | Game date est. |
| GAME_SEQUENCE | character | Game sequence. |
| GAME_ID | character | Unique game identifier. |
| GAME_STATUS_ID | character | Numeric game status identifier. |
| GAME_STATUS_TEXT | character | Game status display text (e.g. 'Final', '4:32 - 4th'). |
| GAMECODE | character | Gamecode. |
| HOME_TEAM_ID | character | Unique identifier for the home team. |
| VISITOR_TEAM_ID | character | Unique identifier for visitor team. |
| SEASON | character | Season identifier (4-digit year or 'YYYY-YY' string). |
| LIVE_PERIOD | character | Live period. |
| LIVE_PC_TIME | character | Time / clock value. |
| NATL_TV_BROADCASTER_ABBREVIATION | character | Natl tv broadcaster abbreviation. |
| HOME_TV_BROADCASTER_ABBREVIATION | character | Home team's tv broadcaster abbreviation. |
| AWAY_TV_BROADCASTER_ABBREVIATION | character | Away team's tv broadcaster abbreviation. |
| LIVE_PERIOD_TIME_BCAST | character | Live period time bcast. |
| ARENA_NAME | character | Arena name. |
| WH_STATUS | character | Wh status. |
| WNBA_COMMISSIONER_FLAG | character | |

**LineScore**

| col_name | types | description |
|---|---|---|
| GAME_DATE_EST | character | Game date est. |
| GAME_SEQUENCE | character | Game sequence. |
| GAME_ID | character | Unique game identifier. |
| TEAM_ID | character | Unique team identifier. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| TEAM_CITY_NAME | character | Team city name. |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| TEAM_WINS_LOSSES | character | Team wins losses. |
| PTS_QTR1 | character | Pts qtr1. |
| PTS_QTR2 | character | Pts qtr2. |
| PTS_QTR3 | character | Pts qtr3. |
| PTS_QTR4 | character | Pts qtr4. |
| PTS_OT1 | character | Pts ot1. |
| PTS_OT2 | character | |
| PTS_OT3 | character | |
| PTS_OT4 | character | |
| PTS_OT5 | character | |
| PTS_OT6 | character | |
| PTS_OT7 | character | |
| PTS_OT8 | character | |
| PTS_OT9 | character | |
| PTS_OT10 | character | |
| PTS | character | Points scored. |
| FG_PCT | character | Field goal percentage (0-1). |
| FT_PCT | character | Free throw percentage (0-1). |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| AST | character | Assists. |
| REB | character | Total rebounds. |
| TOV | character | Turnovers. |

**SeriesStandings**

Columns as documented in the shared wnba_boxscoresummaryv2_season_series_schema table.

**LastMeeting**

Columns as documented in the shared wnba_boxscoresummaryv2_last_meeting_schema table.

**EastConfStandingsByDay**

Columns as documented in the shared wnba_scoreboard_east_conf_standings_by_day_schema table.

**WestConfStandingsByDay**

Columns as documented in the shared wnba_scoreboard_east_conf_standings_by_day_schema table.

**Available**

| col_name | types | description |
|---|---|---|

| | | |
|---|---|---|
| GAME_ID | character | Unique game identifier. |
| PT_AVAILABLE | character | |

**TeamLeaders**

| col_name | types | description |
|---|---|---|
| GAME_ID | character | Unique game identifier. |
| TEAM_ID | character | Unique team identifier. |
| TEAM_CITY | character | Team city or region (e.g. 'Las Vegas'). |
| TEAM_NICKNAME | character | Team nickname. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| PTS_PLAYER_ID | character | |
| PTS_PLAYER_NAME | character | |
| PTS | character | Points scored. |
| REB_PLAYER_ID | character | |
| REB_PLAYER_NAME | character | |
| REB | character | Total rebounds. |
| AST_PLAYER_ID | character | |
| AST_PLAYER_NAME | character | |
| AST | character | Assists. |

**TicketLinks**

**WinProbability**

## Author(s)

Saiem Gilani

## See Also

Other WNBA Schedule Functions: `wnba_schedule()`, `wnba_scoreboardv3()`, `wnba_todays_scoreboard()`

---

`wnba_scoreboardv3` **Get WNBA Stats API Scoreboard V3**

---

## Description

**Get WNBA Stats API Scoreboard V3**

**Get WNBA Stats API Scoreboard V3**

## Usage

```
wnba_scoreboardv3(league_id = "10", game_date = "2022-06-26", ...)
```

### Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| game_date | Game Date |
| ... | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_scoreboardv3(league_id = '10', game_date = '2022-06-26')
```

### Value

Return a tibble with the following scoreboard data:

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Schedule Functions: wnba_schedule(), wnba_scoreboardv2(), wnba_todays_scoreboard()

---

## wnba_shot_zones — Fetch WNBA Play-by-Play and Classify Field-Goal Attempts by Shot Zone

---

### Description

**Fetch WNBA Play-by-Play and Classify Field-Goal Attempts by Shot Zone**

**Fetch WNBA Play-by-Play and Classify Field-Goal Attempts by Shot Zone**

### Usage

```
wnba_shot_zones(game_id, ...)
```

### Arguments

| | |
|---|---|
| game_id | A length-1 character string — the ten-character WNBA Stats game identifier (e.g. "1022400003" for a 2024 WNBA game). |
| ... | Reserved for future keyword arguments (currently ignored). |

### Details

```
df <- wnba_shot_zones(game_id = "1022400003")
print(table(df$shot_zone, useNA = "ifany"))
```

## Value

Returns a `data.frame` containing the full V3 play-by-play for `game_id` with an appended `shot_zone` column:

| col_name | types | description |
|---|---|---|
| event_type | character | Play event type code. `"1"` = MadeShot, `"2"` = MissedShot, `"3"` = FreeThrow, etc. |
| shot_value | integer | Point value of the shot attempt (2 or 3); 0 for non-FGA events. |
| x_legacy | numeric | Horizontal court coordinate (legacy units). Negative = left side of the court. |
| y_legacy | numeric | Vertical court coordinate (legacy units). Values near 0 = baseline. |
| shot_distance | numeric | Distance from the basket in feet. |
| shot_zone | character | Shot-zone classification. One of `"corner_3"`, `"above_the_break_3"`, `"restricted_area"`, `"in` |

All other columns from the V3 PBP schema (`wnba_pbp()`) are also present. Returns a 0-row frame when the game cannot be fetched (never-raise).

**Shot zone definitions (pbpstats-core aligned):**

- `"corner_3"` — three-point attempt with `abs(x_legacy) >= 220` and `y_legacy <= 87.5` (baseline corner area).
- `"above_the_break_3"` — all other three-point attempts.
- `"restricted_area"` — two-point attempt within 4 ft of the basket.
- `"in_the_paint_non_ra"` — two-point attempt 4–8 ft from the basket and within 80 legacy units of the paint center (`abs(x_legacy) <= 80`).
- `"mid_range"` — all remaining two-point attempts.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Lineup Functions: `wnba_gamerotation()`, `wnba_leaguedashlineups()`, `wnba_leaguelineupviz()`, `wnba_possession_lineups()`, `wnba_rapm()`, `wnba_shotchartlineupdetail()`, `wnba_teamdashlineups()`

## Examples

```
try({
  df <- wnba_shot_zones(game_id = "1022400003")
  print(df[!is.na(df$shot_zone), c("shot_distance", "shot_zone")])
})
```

`wnba_shotchartdetail` **Get WNBA Stats API Shot Chart Detail**

## Description

**Get WNBA Stats API Shot Chart Detail**

**Get WNBA Stats API Shot Chart Detail**

## Usage

```
wnba_shotchartdetail(
  context_measure = "FGA",
  date_from = "",
  date_to = "",
  game_id = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  period = 0,
  player_id = "1628932",
  player_position = "",
  rookie_year = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  team_id = 0,
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| `context_measure` | context_measure |
| `date_from` | date_from |
| `date_to` | date_to |
| `game_id` | game_id |
| `game_segment` | game_segment |
| `last_n_games` | last_n_games |
| `league_id` | league_id |

| | |
|---|---|
| location | location |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| period | period |
| player_id | player_id |
| player_position | player_position |
| rookie_year | rookie_year |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_shotchartdetail(league_id = '10', player_id = '1628932', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: LeagueAverages, Shot_Chart_Detail

**Shot_Chart_Detail**

| col_name | types | description |
|---|---|---|
| GRID_TYPE | character | |
| GAME_ID | character | Unique game identifier. |
| GAME_EVENT_ID | character | Unique identifier for game event. |
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |
| TEAM_ID | character | Unique team identifier. |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| PERIOD | character | Period of the game (1-4 quarters; 5+ for OT). |
| MINUTES_REMAINING | character | Minutes remaining. |
| SECONDS_REMAINING | character | Seconds remaining in the period. |
| EVENT_TYPE | character | Event / play type code (V2 PBP). |
| ACTION_TYPE | character | Action type label (e.g. 'Made Shot', 'Substitution'). |
| SHOT_TYPE | character | Shot type label (e.g. 'Jump Shot', 'Layup'). |
| SHOT_ZONE_BASIC | character | Shot zone (e.g. 'Restricted Area', 'Mid-Range', 'Above the Break 3'). |
| SHOT_ZONE_AREA | character | Shot zone area ('Left Side', 'Right Side', 'Center'). |
| SHOT_ZONE_RANGE | character | Shot zone range ('Less Than 8 ft.', '8-16 ft.', '16-24 ft.', etc.). |

| | | |
|---|---|---|
| SHOT_DISTANCE | character | Shot distance from the basket, in feet. |
| LOC_X | character | X coordinate on the court (units of inches; 0 = basket center). |
| LOC_Y | character | Y coordinate on the court (units of inches; baseline at 0). |
| SHOT_ATTEMPTED_FLAG | character | 1 if a shot was attempted on this event. |
| SHOT_MADE_FLAG | character | 1 if the shot was made; 0 if missed. |
| GAME_DATE | character | Game date (YYYY-MM-DD). |
| HTM | character | |
| VTM | character | |

**LeagueAverages**

Columns as documented in the shared wnba_shotchartdetail_league_averages_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Shooting Functions: `wnba_leaguedashplayershotlocations()`, `wnba_leaguedashteamshotlocations()`, `wnba_playerdashboardbyclutch()`, `wnba_shotchartleaguewide()`, `wnba_shotchartlineupdetail()`, `wnba_teamdashboardbyclutch()`

---

`wnba_shotchartleaguewide`

# Get WNBA Stats API Shot Chart League-Wide

---

## Description

**Get WNBA Stats API Shot Chart League-Wide**

**Get WNBA Stats API Shot Chart League-Wide**

## Usage

```
wnba_shotchartleaguewide(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | League - default: '10'. Other options include '00': NBA, '20': G-League |
| `season` | season |
| `...` | Additional arguments passed to an underlying function like httr. |

### Details

```
wnba_shotchartleaguewide(league_id = '10', season = most_recent_wnba_season() - 1)
```

### Value

Return a named list of data frames: League_Wide

**League_Wide**

Columns as documented in the shared wnba_shotchartdetail_league_averages_schema table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Shooting Functions: wnba_leaguedashplayershotlocations(), wnba_leaguedashteamshotlocations(), wnba_playerdashboardbyclutch(), wnba_shotchartdetail(), wnba_shotchartlineupdetail(), wnba_teamdashboardbyclutch()

Other WNBA League Functions: wnba_leaguedashlineups(), wnba_leaguedashplayerbiostats(), wnba_leaguedashplayerclutch(), wnba_leaguedashplayershotlocations(), wnba_leaguedashplayerstats(), wnba_leaguedashteamclutch(), wnba_leaguedashteamshotlocations(), wnba_leaguedashteamstats(), wnba_leaguegamefinder(), wnba_leaguegamelog(), wnba_leaguelineupviz(), wnba_leagueplayerondetails(), wnba_leagueseasonmatchups(), wnba_leaguestandingsv3()

---

`wnba_shotchartlineupdetail`

## Get WNBA Stats API Shot Chart for Lineups

---

### Description

**Get WNBA Stats API Shot Chart for Lineups**

**Get WNBA Stats API Shot Chart for Lineups**

### Usage

```
wnba_shotchartlineupdetail(
  ahead_behind = "",
  cfid = "",
  cfparams = "",
  clutch_time = "",
  conference = "",
  context_filter = "",
  context_measure = "FGA",
  date_from = "",
  date_to = "",
  division = "",
```

```
  end_period = "10",
  end_range = "28800",
  group_id = "-1628899-1629481-1630096-1631019-1642784-",
  game_event_id = "",
  game_id = "",
  game_segment = "",
  group_mode = "",
  group_quantity = "5",
  last_n_games = "0",
  league_id = "10",
  location = "",
  month = "0",
  on_off = "",
  opp_player_id = "",
  opponent_team_id = "0",
  outcome = "",
  po_round = "0",
  period = "0",
  player_id = "0",
  player_id1 = "",
  player_id2 = "",
  player_id3 = "",
  player_id4 = "",
  player_id5 = "",
  player_position = "",
  point_diff = "",
  position = "",
  range_type = "0",
  rookie_year = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  start_period = "1",
  start_range = "0",
  starter_bench = "",
  team_id = "1611661321",
  vs_conference = "",
  vs_division = "",
  vs_player_id1 = "",
  vs_player_id2 = "",
  vs_player_id3 = "",
  vs_player_id4 = "",
  vs_player_id5 = "",
  vs_team_id = "",
  ...
)
```

## Arguments

| | |
|---|---|
| ahead_behind | ahead_behind |
| cfid | cfid |
| cfparams | cfparams |
| clutch_time | clutch_time |
| conference | conference |
| context_filter | context_filter |
| context_measure | context_measure |
| date_from | date_from |
| date_to | date_to |
| division | division |
| end_period | end_period |
| end_range | end_range |
| group_id | group_id |
| game_event_id | game_event_id |
| game_id | game_id |
| game_segment | game_segment |
| group_mode | group_mode |
| group_quantity | group_quantity |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| month | month |
| on_off | on_off |
| opp_player_id | opp_player_id |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| period | period |
| player_id | player_id |
| player_id1 | player_id1 |
| player_id2 | player_id2 |
| player_id3 | player_id3 |
| player_id4 | player_id4 |
| player_id5 | player_id5 |

player_position
: player_position

point_diff
: point_diff

position
: position

range_type
: range_type

rookie_year
: rookie_year

season
: season

season_segment
: season_segment

season_type
: season_type

shot_clock_range
: shot_clock_range

start_period
: start_period

start_range
: start_range

starter_bench
: starter_bench

team_id
: team_id

vs_conference
: vs_conference

vs_division
: vs_division

vs_player_id1
: vs_player_id1

vs_player_id2
: vs_player_id2

vs_player_id3
: vs_player_id3

vs_player_id4
: vs_player_id4

vs_player_id5
: vs_player_id5

vs_team_id
: vs_team_id

...
: Additional arguments passed to an underlying function like httr.

## Details

```
wnba_shotchartlineupdetail(group_id = '-1628899-1629481-1630096-1631019-1642784-', opponent_team_id
```

## Value

Return a named list of data frames: ShotChartLineupDetail, ShotChartLineupLeagueAverage

**ShotChartLineupDetail**

| col_name | types | description |
|---|---|---|
| GRID_TYPE | character | |
| GAME_ID | character | Unique game identifier. |
| GAME_EVENT_ID | character | Unique identifier for game event. |
| GROUP_ID | character | Group identifier (e.g. conference group_id). |
| GROUP_NAME | character | Group name. |
| PLAYER_ID | character | Unique player identifier. |
| PLAYER_NAME | character | Player name. |

| | | |
|---|---|---|
| TEAM_ID | character | Unique team identifier. |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| PERIOD | character | Period of the game (1-4 quarters; 5+ for OT). |
| MINUTES_REMAINING | character | Minutes remaining. |
| SECONDS_REMAINING | character | Seconds remaining in the period. |
| EVENT_TYPE | character | Event / play type code (V2 PBP). |
| ACTION_TYPE | character | Action type label (e.g. 'Made Shot', 'Substitution'). |
| SHOT_TYPE | character | Shot type label (e.g. 'Jump Shot', 'Layup'). |
| SHOT_ZONE_BASIC | character | Shot zone (e.g. 'Restricted Area', 'Mid-Range', 'Above the Break 3'). |
| SHOT_ZONE_AREA | character | Shot zone area ('Left Side', 'Right Side', 'Center'). |
| SHOT_ZONE_RANGE | character | Shot zone range ('Less Than 8 ft.', '8-16 ft.', '16-24 ft.', etc.). |
| SHOT_DISTANCE | character | Shot distance from the basket, in feet. |
| LOC_X | character | X coordinate on the court (units of inches; 0 = basket center). |
| LOC_Y | character | Y coordinate on the court (units of inches; baseline at 0). |
| SHOT_ATTEMPTED_FLAG | character | 1 if a shot was attempted on this event. |
| SHOT_MADE_FLAG | character | 1 if the shot was made; 0 if missed. |
| GAME_DATE | character | Game date (YYYY-MM-DD). |
| HTM | character | |
| VTM | character | |

**ShotChartLineupLeagueAverage**

Columns as documented in the shared wnba_shotchartdetail_league_averages_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Shooting Functions: `wnba_leaguedashplayershotlocations()`, `wnba_leaguedashteamshotlocations()`, `wnba_playerdashboardbyclutch()`, `wnba_shotchartdetail()`, `wnba_shotchartleaguewide()`, `wnba_teamdashboardbyclutch()`

Other WNBA Lineup Functions: `wnba_gamerotation()`, `wnba_leaguedashlineups()`, `wnba_leaguelineupviz()`, `wnba_possession_lineups()`, `wnba_rapm()`, `wnba_shot_zones()`, `wnba_teamdashlineups()`

---

`wnba_stats` **WNBA Stats API Endpoint Overview**

---

## Description

Wrappers around the WNBA Stats backend at `stats.wnba.com/stats/...`. All requests go through `request_with_proxy()` in `utils_wnba_stats.R` which sets the required WNBA-side headers (`Origin: https://stats.wnba.com`, `Referer: https://www.wnba.com/`) and threads a proxy via `getOption("wehoop.proxy")` / env vars / explicit `proxy =` arg.

WNBA-specific differences vs the NBA Stats API: quarters are 10 minutes (not 12), regulation is 40 minutes (not 48), game IDs start with `"10"`, `LeagueID = "10"`.

## Details

**Boxscores (V2):**

Tabular `resultSets[]` responses parsed via `wnba_stats_map_result_sets()`. See `wnba_boxscoreadvancedv2()`, `wnba_boxscoretraditionalv2()`, `wnba_boxscorefourfactorsv2()`, `wnba_boxscoremiscv2()`, `wnba_boxscoreplayertrackv2()`, `wnba_boxscorescoringv2()`, `wnba_boxscoresummaryv2()`, `wnba_boxscoreusagev2()`, `wnba_hustlestatsboxscore()`.

**Boxscores (V3):**

Nested JSON parsed via `purrr::pluck()`. See `wnba_boxscoreadvancedv3()`, `wnba_boxscoretraditionalv3()`, `wnba_boxscorefourfactorsv3()`, `wnba_boxscoremiscv3()`, `wnba_boxscoreplayertrackv3()`, `wnba_boxscorescoringv3()`, `wnba_boxscoreusagev3()`.

**Play-by-play:**

| Function | Purpose |
|---|---|
| `wnba_pbp()` | V2 + V3 PBP (with V3-to-V2 conversion pipeline) |
| `wnba_pbps()` | Multi-game PBP loader |
| `wnba_playbyplayv3()` | V3-only PBP |

**Leaders, standings, league:**

`wnba_leagueleaders()`, `wnba_leaguestandingsv3()`, `wnba_homepageleaders()`, `wnba_leaderstiles()`.

**Player + team season + game stats:**

Full per-mode / per-split families. Search the reference index for `wnba_playerdashboard*`, `wnba_teamdashboard*`, `wnba_leaguedashplayerstats()`, `wnba_leaguedashteamstats()`, `wnba_playergamelogs()`, `wnba_teamgamelogs()`, `wnba_playerprofile*`.

**Schedule + scoreboard:**

`wnba_schedule()`, `wnba_scoreboard()`, `wnba_scoreboardv2()`, `wnba_scoreboardv3()`, `wnba_todays_scoreboard()`

**Draft + franchise + reference:**

`wnba_draftboard()`, `wnba_drafthistory()`, `wnba_draftcombinestats()`, `wnba_franchisehistory()`, `wnba_franchiseleaders()`.

---

# `wnba_team_crosswalk` Get the WNBA cross-source team crosswalk

---

## Description

Build a wide, one-row-per-team-per-season crosswalk linking ESPN, the WNBA Stats API, and Fox Sports team identities, keyed on `espn_team_id`. Yahoo columns are placeholders (NA) until that source is implemented.

## Usage

```
wnba_team_crosswalk(season = most_recent_wnba_season(), .schedule = NULL)
```

### Arguments

| | |
|---|---|
| season | Season year (numeric). Defaults to the most recent WNBA season. |
| .schedule | Internal. An already-fetched wnba_schedule() frame, supplied by wnba_schedule_crosswalk() to avoid a duplicate CDN request. Leave NULL. |

### Value

A wehoop_data tibble, one row per team:

See the HTML help or pkgdown reference for the column table.

### See Also

Other WNBA Crosswalk Functions: load_wnba_team_crosswalk(), wnba_player_crosswalk(), wnba_schedule_crosswalk()

### Examples

```
try(wnba_team_crosswalk(season = 2024))
```

---

## wnba_teamdashboardbyclutch

## Get WNBA Stats API Team Dashboard by Clutch Splits

---

### Description

**Get WNBA Stats API Team Dashboard by Clutch Splits**

**Get WNBA Stats API Team Dashboard by Clutch Splits**

**Get WNBA Stats API Team Dashboard by Game Splits**

**Get WNBA Stats API Team Dashboard by General Splits**

**Get WNBA Stats API Team Dashboard by Last N Games**

**Get WNBA Stats API Team Dashboard by Opponent**

**Get WNBA Stats API Team Dashboard by Shooting Splits**

**Get WNBA Stats API Team Dashboard by Team Performance**

**Get WNBA Stats API Team Dashboard Year over Year**

### Usage

```
wnba_teamdashboardbyclutch(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
```

```
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbygamesplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)
```

```
wnba_teamdashboardbygeneralsplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbylastngames(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
```

```
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbyopponent(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbyshootingsplits(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
```

```
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbyteamperformance(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamdashboardbyyearoveryear(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
```

```
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| plus_minus | plus_minus |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |

| | |
|---|---|
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamdashboardbyclutch(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbygamesplits(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbygeneralsplits(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbylastngames(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbyopponent(team_id = '1611661328', season = most_recent_wnba_season() - 1)
```

Team Dashboard by Shooting Splits

```
wnba_teamdashboardbyshootingsplits(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbyteamperformance(team_id = '1611661328', season = most_recent_wnba_season() - 1)

wnba_teamdashboardbyyearoveryear(team_id = '1611661328', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: Last10Sec3Point2TeamDashboard, Last10Sec3PointTeamDashboard, Last1Min5PointTeamDashboard, Last1MinPlusMinus5PointTeamDashboard, Last30Sec3Point2TeamDashboard, Last30Sec3PointTeamDashboard, Last3Min5PointTeamDashboard, Last3MinPlusMinus5PointTeamDashboard, Last5Min5PointTeamDashboard, Last5MinPlusMinus5PointTeamDashboard, OverallTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last5Min5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last3Min5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last1Min5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last30Sec3PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last10Sec3PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last5MinPlusMinus5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last3MinPlusMinus5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last1MinPlusMinus5PointTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last30Sec3Point2TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last10Sec3Point2TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

Return a named list of data frames: ByActualMarginTeamDashboard, ByHalfTeamDashboard, ByPeriodTeamDashboard, ByScoreMarginTeamDashboard, OverallTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ByHalfTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ByPeriodTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ByScoreMarginTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ByActualMarginTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

Return a named list of data frames: DaysRestTeamDashboard, LocationTeamDashboard, MonthTeamDashboard, OverallTeamDashboard, PrePostAllStarTeamDashboard, WinsLossesTeamDashboard

**OverallTeamDashboard**

See the HTML help or pkgdown reference for the column table.

**LocationTeamDashboard**

See the HTML help or pkgdown reference for the column table.

**WinsLossesTeamDashboard**

See the HTML help or pkgdown reference for the column table.

**MonthTeamDashboard**

See the HTML help or pkgdown reference for the column table.

**PrePostAllStarTeamDashboard**

See the HTML help or pkgdown reference for the column table.

**DaysRestTeamDashboard**

See the HTML help or pkgdown reference for the column table.

Return a named list of data frames: GameNumberTeamDashboard, Last10TeamDashboard, Last15TeamDashboard, Last20TeamDashboard, Last5TeamDashboard, OverallTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last5TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last10TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last15TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**Last20TeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**GameNumberTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

Return a named list of data frames: ConferenceTeamDashboard, DivisionTeamDashboard, OpponentTeamDashboard, OverallTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ConferenceTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**DivisionTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**OpponentTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

Return a named list of data frames: AssistedBy, AssitedShotTeamDashboard, OverallTeamDashboard, Shot5FTTeamDashboard, Shot8FTTeamDashboard, ShotAreaTeamDashboard, ShotTypeTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**Shot5FTTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**Shot8FTTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**ShotAreaTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**AssitedShotTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**ShotTypeTeamDashboard**

Columns as documented in the shared wnba_shooting_splits_schema table.

**AssistedBy**

Columns as documented in the shared wnba_shooting_splits_assisted_by_schema table.

Return a named list of data frames: OverallTeamDashboard, PointsScoredTeamDashboard, PontsAgainstTeamDashboard, ScoreDifferentialTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ScoreDifferentialTeamDashboard**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE_ORDER | character | |
| GROUP_VALUE | character | |
| GROUP_VALUE_2 | character | |
| GP | character | Games played. |
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |
| FGM_RANK | character | |
| FGA_RANK | character | |
| FG_PCT_RANK | character | |

| | |
|---|---|
| FG3M_RANK | character |
| FG3A_RANK | character |
| FG3_PCT_RANK | character |
| FTM_RANK | character |
| FTA_RANK | character |
| FT_PCT_RANK | character |
| OREB_RANK | character |
| DREB_RANK | character |
| REB_RANK | character |
| AST_RANK | character |
| TOV_RANK | character |
| STL_RANK | character |
| BLK_RANK | character |
| BLKA_RANK | character |
| PF_RANK | character |
| PFD_RANK | character |
| PTS_RANK | character |
| PLUS_MINUS_RANK | character |

**PointsScoredTeamDashboard**

Same columns as the **ScoreDifferentialTeamDashboard** table above.

**PontsAgainstTeamDashboard**

Same columns as the **ScoreDifferentialTeamDashboard** table above.

Return a named list of data frames: ByYearTeamDashboard, OverallTeamDashboard

**OverallTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

**ByYearTeamDashboard**

Columns as documented in the shared wnba_team_dashboard_schema table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: `wnba_leaguedashteamstats()`, `wnba_teamdashlineups()`, `wnba_teamdetails()`, `wnba_teamestimatedmetrics()`, `wnba_teamgamelog()`, `wnba_teamgamelogs()`, `wnba_teamgamestreakfinder()`, `wnba_teamhistoricalleaders()`, `wnba_teaminfocommon()`, `wnba_teamplayeronoffdetails()`, `wnba_teams()`, `wnba_teamvsplayer()`, `wnba_teamyearbyyearstats()`

Other WNBA Clutch Functions: `wnba_leaguedashteamclutch()`, `wnba_playerdashboardbyclutch()`

Other WNBA Shooting Functions: `wnba_leaguedashplayershotlocations()`, `wnba_leaguedashteamshotlocations()`, `wnba_playerdashboardbyclutch()`, `wnba_shotchartdetail()`, `wnba_shotchartleaguewide()`, `wnba_shotchartlineupdetail()`

wnba_teamdashlineups **Get WNBA Stats API Team Dashboard - Lineups**

## Description

**Get WNBA Stats API Team Dashboard - Lineups**

**Get WNBA Stats API Team Dashboard - Lineups**

## Usage

```
wnba_teamdashlineups(
  date_from = "",
  date_to = "",
  game_id = "",
  game_segment = "",
  group_quantity = 5,
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_id | game_id |
| game_segment | game_segment |

| | |
|---|---|
| group_quantity | group_quantity |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| plus_minus | plus_minus |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamdashlineups(team_id = '1611661328', season = most_recent_wnba_season() - 1)
```

## Value

Return a named list of data frames: Lineups, Overall

### Overall

Columns as documented in the shared wnba_team_summary_overall_schema table.

### Lineups

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_ID | character | Group identifier (e.g. conference group_id). |
| GROUP_NAME | character | Group name. |
| GP | character | Games played. |

| | | |
|---|---|---|
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |
| FGM_RANK | character | |
| FGA_RANK | character | |
| FG_PCT_RANK | character | |
| FG3M_RANK | character | |
| FG3A_RANK | character | |
| FG3_PCT_RANK | character | |
| FTM_RANK | character | |
| FTA_RANK | character | |
| FT_PCT_RANK | character | |
| OREB_RANK | character | |
| DREB_RANK | character | |
| REB_RANK | character | |
| AST_RANK | character | |
| TOV_RANK | character | |
| STL_RANK | character | |
| BLK_RANK | character | |
| BLKA_RANK | character | |
| PF_RANK | character | |

| | |
|---|---|
| PFD_RANK | character |
| PTS_RANK | character |
| PLUS_MINUS_RANK | character |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

Other WNBA Lineup Functions: wnba_gamerotation(), wnba_leaguedashlineups(), wnba_leaguelineupviz(), wnba_possession_lineups(), wnba_rapm(), wnba_shot_zones(), wnba_shotchartlineupdetail()

---

# wnba_teamdetails Get WNBA Stats API Team Details

---

## Description

**Get WNBA Stats API Team Details**

**Get WNBA Stats API Team Details**

## Usage

```
wnba_teamdetails(team_id = "1611661328", ...)
```

## Arguments

| | |
|---|---|
| team_id | Team ID |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamdetails(team_id = '1611661328')
```

## Value

Return a named list of data frames: TeamAwardsChampionships, TeamAwardsConf, TeamAwardsDiv, TeamBackground, TeamHistory, TeamHof, TeamRetired, TeamSocialSites

**TeamBackground**

See the HTML help or pkgdown reference for the column table.

**TeamHistory**

| col_name | types | description |
|---|---|---|
| TEAM_ID | character | Unique team identifier. |
| CITY | character | City. |
| NICKNAME | character | Team or athlete nickname. |
| YEARFOUNDED | character | |
| YEARACTIVETILL | character | |

**TeamSocialSites**

| col_name | types | description |
|---|---|---|
| ACCOUNTTYPE | character | |
| WEBSITE_LINK | character | |

**TeamAwardsChampionships**

| col_name | types | description |
|---|---|---|
| YEARAWARDED | character | |
| OPPOSITETEAM | character | |

**TeamAwardsConf**

| col_name | types | description |
|---|---|---|
| YEARAWARDED | integer | |
| OPPOSITETEAM | integer | |

**TeamAwardsDiv**

Same columns as the **TeamAwardsConf** table above.

**TeamHof**

| col_name | types | description |
|---|---|---|
| PLAYERID | character | |
| PLAYER | character | |
| POSITION | character | Listed roster position (G, F, C, etc.). |
| JERSEY | character | Jersey number worn by the player. |
| SEASONSWITHTEAM | character | |
| YEAR | character | 4-digit year. |

**TeamRetired**

Same columns as the **TeamHof** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

`wnba_teamestimatedmetrics`

# Get WNBA Stats API Team Estimated Metrics

---

## Description

**Get WNBA Stats API Team Estimated Metrics**

**Get WNBA Stats API Team Estimated Metrics**

## Usage

```
wnba_teamestimatedmetrics(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| `season` | Season - format 2020-21 |
| `season_type` | Season Type - Regular Season, Playoffs, All-Star |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamestimatedmetrics()
```

## Value

Return a named list of data frames: TeamEstimatedMetrics

**TeamEstimatedMetrics**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

`wnba_teamgamelog` **Get WNBA Stats API Team Game Log**

---

## Description

**Get WNBA Stats API Team Game Log**

**Get WNBA Stats API Team Game Log**

## Usage

```
wnba_teamgamelog(
  date_from = "",
  date_to = "",
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  team_id = "1611661328",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| league_id | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | Team ID |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamgamelog(team_id = '1611661328')
```

## Value

Return a named list of data frames: TeamGameLog

**TeamGameLog**

See the HTML help or pkgdown reference for the column table.

### Author(s)

Saiem Gilani

### See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

## wnba_teamgamelogs Get WNBA Stats API Team Game Logs

---

### Description

**Get WNBA Stats API Team Game Logs**

**Get WNBA Stats API Team Game Logs**

### Usage

```
wnba_teamgamelogs(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  player_id = "",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| player_id | Player ID |
| season | Season - format 2020-21 |
| season_segment | season_segment |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

Teams Game Log

```
wnba_teamgamelogs(team_id = '1611661328')
```

## Value

Return a named list of data frames: TeamGameLogs

**TeamGameLogs**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

wnba_teamgamestreakfinder

**Get WNBA Stats API Team Game Streak Finder**

---

## Description

**Get WNBA Stats API Team Game Streak Finder**

**Get WNBA Stats API Team Game Streak Finder**

## Usage

```
wnba_teamgamestreakfinder(
  active_streaks_only = "",
  active_teams_only = "",
  btr_opp_ast = "",
  btr_opp_blk = "",
  btr_opp_dreb = "",
  btr_opp_fg3a = "",
  btr_opp_fg3m = "",
  btr_opp_fg3_pct = "",
  btr_opp_fga = "",
  btr_opp_fgm = "",
  btr_opp_fg_pct = "",
  btr_opp_fta = "",
  btr_opp_ftm = "",
  btr_opp_ft_pct = "",
  btr_opp_oreb = "",
  btr_opp_pf = "",
  btr_opp_pts = "",
  btr_opp_pts2nd_chance = "",
  btr_opp_pts_fb = "",
  btr_opp_pts_off_tov = "",
  btr_opp_pts_paint = "",
  btr_opp_reb = "",
  btr_opp_stl = "",
  btr_opp_tov = "",
  conference = "",
  date_from = "",
  date_to = "",
  division = "",
```

```
    et_ast = "",
    et_blk = "",
    et_dd = "",
    et_dreb = "",
    et_fg3a = "",
    et_fg3m = "",
    et_fg3_pct = "",
    et_fga = "",
    et_fgm = "",
    et_fg_pct = "",
    et_fta = "",
    et_ftm = "",
    et_ft_pct = "",
    et_minutes = "",
    eq_opp_pts2nd_chance = "",
    eq_opp_pts_fb = "",
    eq_opp_pts_off_tov = "",
    eq_opp_pts_paint = "",
    et_oreb = "",
    et_pf = "",
    et_pts = "",
    eq_pts2nd_chance = "",
    eq_pts_fb = "",
    eq_pts_off_tov = "",
    eq_pts_paint = "",
    et_reb = "",
    et_stl = "",
    et_td = "",
    et_tov = "",
    game_id = "",
    gt_ast = "",
    gt_blk = "",
    gt_dd = "",
    gt_dreb = "",
    gt_fg3a = "",
    gt_fg3m = "",
    gt_fg3_pct = "",
    gt_fga = "",
    gt_fgm = "",
    gt_fg_pct = "",
    gt_fta = "",
    gt_ftm = "",
    gt_ft_pct = "",
    gt_minutes = "",
    gt_opp_ast = "",
    gt_opp_blk = "",
    gt_opp_dreb = "",
    gt_opp_fg3a = "",
```

```
    gt_opp_fg3m = "",
    gt_opp_fg3_pct = "",
    gt_opp_fga = "",
    gt_opp_fgm = "",
    gt_opp_fg_pct = "",
    gt_opp_fta = "",
    gt_opp_ftm = "",
    gt_opp_ft_pct = "",
    gt_opp_oreb = "",
    gt_opp_pf = "",
    gt_opp_pts = "",
    gt_opp_pts2nd_chance = "",
    gt_opp_pts_fb = "",
    gt_opp_pts_off_tov = "",
    gt_opp_pts_paint = "",
    gt_opp_reb = "",
    gt_opp_stl = "",
    gt_opp_tov = "",
    gt_oreb = "",
    gt_pf = "",
    gt_pts = "",
    gt_pts2nd_chance = "",
    gt_pts_fb = "",
    gt_pts_off_tov = "",
    gt_pts_paint = "",
    gt_reb = "",
    gt_stl = "",
    gt_td = "",
    gt_tov = "",
    lstreak = "",
    league_id = "10",
    location = "",
    lt_ast = "",
    lt_blk = "",
    lt_dd = "",
    lt_dreb = "",
    lt_fg3a = "",
    lt_fg3m = "",
    lt_fg3_pct = "",
    lt_fga = "",
    lt_fgm = "",
    lt_fg_pct = "",
    lt_fta = "",
    lt_ftm = "",
    lt_ft_pct = "",
    lt_minutes = "",
    lt_opp_ast = "",
    lt_opp_blk = "",
```

```
    lt_opp_dreb = "",
    lt_opp_fg3a = "",
    lt_opp_fg3m = "",
    lt_opp_fg3_pct = "",
    lt_opp_fga = "",
    lt_opp_fgm = "",
    lt_opp_fg_pct = "",
    lt_opp_fta = "",
    lt_opp_ftm = "",
    lt_opp_ft_pct = "",
    lt_opp_oreb = "",
    lt_opp_pf = "",
    lt_opp_pts = "",
    lt_opp_pts2nd_chance = "",
    lt_opp_pts_fb = "",
    lt_opp_pts_off_tov = "",
    lt_opp_pts_paint = "",
    lt_opp_reb = "",
    lt_opp_stl = "",
    lt_opp_tov = "",
    lt_oreb = "",
    lt_pf = "",
    lt_pts = "",
    lt_pts2nd_chance = "",
    lt_pts_fb = "",
    lt_pts_off_tov = "",
    lt_pts_paint = "",
    lt_reb = "",
    lt_stl = "",
    lt_td = "",
    lt_tov = "",
    min_games = "",
    outcome = "",
    po_round = "",
    season = most_recent_wnba_season() - 1,
    season_segment = "",
    season_type = "Regular Season",
    team_id = "",
    vs_conference = "",
    vs_division = "",
    vs_team_id = "",
    wstreak = "",
    wrs_opp_ast = "",
    wrs_opp_blk = "",
    wrs_opp_dreb = "",
    wrs_opp_fg3a = "",
    wrs_opp_fg3m = "",
    wrs_opp_fg3_pct = "",
```

```
  wrs_opp_fga = "",
  wrs_opp_fgm = "",
  wrs_opp_fg_pct = "",
  wrs_opp_fta = "",
  wrs_opp_ftm = "",
  wrs_opp_ft_pct = "",
  wrs_opp_oreb = "",
  wrs_opp_pf = "",
  wrs_opp_pts = "",
  wrs_opp_pts2nd_chance = "",
  wrs_opp_pts_fb = "",
  wrs_opp_pts_off_tov = "",
  wrs_opp_pts_paint = "",
  wrs_opp_reb = "",
  wrs_opp_stl = "",
  wrs_opp_tov = "",
  ...
)
```

## Arguments

| | |
|---|---|
| active_streaks_only | active_streaks_only |
| active_teams_only | active_teams_only |
| btr_opp_ast | btr_opp_ast |
| btr_opp_blk | btr_opp_blk |
| btr_opp_dreb | btr_opp_dreb |
| btr_opp_fg3a | btr_opp_fg3a |
| btr_opp_fg3m | btr_opp_fg3m |
| btr_opp_fg3_pct | btr_opp_fg3_pct |
| btr_opp_fga | btr_opp_fga |
| btr_opp_fgm | btr_opp_fgm |
| btr_opp_fg_pct | btr_opp_fg_pct |
| btr_opp_fta | btr_opp_fta |
| btr_opp_ftm | btr_opp_ftm |
| btr_opp_ft_pct | btr_opp_ft_pct |
| btr_opp_oreb | btr_opp_oreb |
| btr_opp_pf | btr_opp_pf |
| btr_opp_pts | btr_opp_pts |
| btr_opp_pts2nd_chance | btr_opp_pts2nd_chance |
| btr_opp_pts_fb | btr_opp_pts_fb |

btr_opp_pts_off_tov
: btr_opp_pts_off_tov

btr_opp_pts_paint
: btr_opp_pts_paint

btr_opp_reb
: btr_opp_reb

btr_opp_stl
: btr_opp_stl

btr_opp_tov
: btr_opp_tov

conference
: conference

date_from
: date_from

date_to
: date_to

division
: division

et_ast
: et_ast

et_blk
: et_blk

et_dd
: et_dd

et_dreb
: et_dreb

et_fg3a
: et_fg3a

et_fg3m
: et_fg3m

et_fg3_pct
: et_fg3_pct

et_fga
: et_fga

et_fgm
: et_fgm

et_fg_pct
: et_fg_pct

et_fta
: et_fta

et_ftm
: et_ftm

et_ft_pct
: et_ft_pct

et_minutes
: et_minutes

eq_opp_pts2nd_chance
: eq_opp_pts2nd_chance

eq_opp_pts_fb
: eq_opp_pts_fb

eq_opp_pts_off_tov
: eq_opp_pts_off_tov

eq_opp_pts_paint
: eq_opp_pts_paint

et_oreb
: et_oreb

et_pf
: et_pf

et_pts
: et_pts

eq_pts2nd_chance
: eq_pts2nd_chance

eq_pts_fb
: eq_pts_fb

eq_pts_off_tov
: eq_pts_off_tov

| | |
|---|---|
| eq_pts_paint | eq_pts_paint |
| et_reb | et_reb |
| et_stl | et_stl |
| et_td | et_td |
| et_tov | et_tov |
| game_id | game_id |
| gt_ast | gt_ast |
| gt_blk | gt_blk |
| gt_dd | gt_dd |
| gt_dreb | gt_dreb |
| gt_fg3a | gt_fg3a |
| gt_fg3m | gt_fg3m |
| gt_fg3_pct | gt_fg3_pct |
| gt_fga | gt_fga |
| gt_fgm | gt_fgm |
| gt_fg_pct | gt_fg_pct |
| gt_fta | gt_fta |
| gt_ftm | gt_ftm |
| gt_ft_pct | gt_ft_pct |
| gt_minutes | gt_minutes |
| gt_opp_ast | gt_opp_ast |
| gt_opp_blk | gt_opp_blk |
| gt_opp_dreb | gt_opp_dreb |
| gt_opp_fg3a | gt_opp_fg3a |
| gt_opp_fg3m | gt_opp_fg3m |
| gt_opp_fg3_pct | gt_opp_fg3_pct |
| gt_opp_fga | gt_opp_fga |
| gt_opp_fgm | gt_opp_fgm |
| gt_opp_fg_pct | gt_opp_fg_pct |
| gt_opp_fta | gt_opp_fta |
| gt_opp_ftm | gt_opp_ftm |
| gt_opp_ft_pct | gt_opp_ft_pct |
| gt_opp_oreb | gt_opp_oreb |
| gt_opp_pf | gt_opp_pf |
| gt_opp_pts | gt_opp_pts |
| gt_opp_pts2nd_chance | gt_opp_pts2nd_chance |

| | |
|---|---|
| gt_opp_pts_fb | gt_opp_pts_fb |
| gt_opp_pts_off_tov | gt_opp_pts_off_tov |
| gt_opp_pts_paint | gt_opp_pts_paint |
| gt_opp_reb | gt_opp_reb |
| gt_opp_stl | gt_opp_stl |
| gt_opp_tov | gt_opp_tov |
| gt_oreb | gt_oreb |
| gt_pf | gt_pf |
| gt_pts | gt_pts |
| gt_pts2nd_chance | gt_pts2nd_chance |
| gt_pts_fb | gt_pts_fb |
| gt_pts_off_tov | gt_pts_off_tov |
| gt_pts_paint | gt_pts_paint |
| gt_reb | gt_reb |
| gt_stl | gt_stl |
| gt_td | gt_td |
| gt_tov | gt_tov |
| lstreak | lstreak |
| league_id | league_id |
| location | location |
| lt_ast | lt_ast |
| lt_blk | lt_blk |
| lt_dd | lt_dd |
| lt_dreb | lt_dreb |
| lt_fg3a | lt_fg3a |
| lt_fg3m | lt_fg3m |
| lt_fg3_pct | lt_fg3_pct |
| lt_fga | lt_fga |
| lt_fgm | lt_fgm |
| lt_fg_pct | lt_fg_pct |
| lt_fta | lt_fta |
| lt_ftm | lt_ftm |
| lt_ft_pct | lt_ft_pct |
| lt_minutes | lt_minutes |
| lt_opp_ast | lt_opp_ast |

| | |
|---|---|
| lt_opp_blk | lt_opp_blk |
| lt_opp_dreb | lt_opp_dreb |
| lt_opp_fg3a | lt_opp_fg3a |
| lt_opp_fg3m | lt_opp_fg3m |
| lt_opp_fg3_pct | lt_opp_fg3_pct |
| lt_opp_fga | lt_opp_fga |
| lt_opp_fgm | lt_opp_fgm |
| lt_opp_fg_pct | lt_opp_fg_pct |
| lt_opp_fta | lt_opp_fta |
| lt_opp_ftm | lt_opp_ftm |
| lt_opp_ft_pct | lt_opp_ft_pct |
| lt_opp_oreb | lt_opp_oreb |
| lt_opp_pf | lt_opp_pf |
| lt_opp_pts | lt_opp_pts |
| lt_opp_pts2nd_chance | lt_opp_pts2nd_chance |
| lt_opp_pts_fb | lt_opp_pts_fb |
| lt_opp_pts_off_tov | lt_opp_pts_off_tov |
| lt_opp_pts_paint | lt_opp_pts_paint |
| lt_opp_reb | lt_opp_reb |
| lt_opp_stl | lt_opp_stl |
| lt_opp_tov | lt_opp_tov |
| lt_oreb | lt_oreb |
| lt_pf | lt_pf |
| lt_pts | lt_pts |
| lt_pts2nd_chance | lt_pts2nd_chance |
| lt_pts_fb | lt_pts_fb |
| lt_pts_off_tov | lt_pts_off_tov |
| lt_pts_paint | lt_pts_paint |
| lt_reb | lt_reb |
| lt_stl | lt_stl |
| lt_td | lt_td |
| lt_tov | lt_tov |
| min_games | min_games |
| outcome | outcome |

| | |
|---|---|
| po_round | po_round |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_team_id | vs_team_id |
| wstreak | wstreak |
| wrs_opp_ast | wrs_opp_ast |
| wrs_opp_blk | wrs_opp_blk |
| wrs_opp_dreb | wrs_opp_dreb |
| wrs_opp_fg3a | wrs_opp_fg3a |
| wrs_opp_fg3m | wrs_opp_fg3m |
| wrs_opp_fg3_pct | wrs_opp_fg3_pct |
| wrs_opp_fga | wrs_opp_fga |
| wrs_opp_fgm | wrs_opp_fgm |
| wrs_opp_fg_pct | wrs_opp_fg_pct |
| wrs_opp_fta | wrs_opp_fta |
| wrs_opp_ftm | wrs_opp_ftm |
| wrs_opp_ft_pct | wrs_opp_ft_pct |
| wrs_opp_oreb | wrs_opp_oreb |
| wrs_opp_pf | wrs_opp_pf |
| wrs_opp_pts | wrs_opp_pts |
| wrs_opp_pts2nd_chance | wrs_opp_pts2nd_chance |
| wrs_opp_pts_fb | wrs_opp_pts_fb |
| wrs_opp_pts_off_tov | wrs_opp_pts_off_tov |
| wrs_opp_pts_paint | wrs_opp_pts_paint |
| wrs_opp_reb | wrs_opp_reb |
| wrs_opp_stl | wrs_opp_stl |
| wrs_opp_tov | wrs_opp_tov |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamgamestreakfinder()
```

## Value

Return a named list of data frames: TeamGameStreakFinderParametersResults

**TeamGameStreakFinderParametersResults**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails() wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

Other WNBA Game Finder Functions: wnba_leaguegamefinder(), wnba_playergamestreakfinder()

---

`wnba_teamhistoricalleaders`

# Get WNBA Stats API Team Historical Leaders

---

## Description

**Get WNBA Stats API Team Historical Leaders**

**Get WNBA Stats API Team Historical Leaders**

## Usage

```
wnba_teamhistoricalleaders(
  league_id = "10",
  season_id = "22022",
  team_id = "1611661328",
  ...
)
```

## Arguments

| | |
|---|---|
| `league_id` | league_id |
| `season_id` | season_id |
| `team_id` | team_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamhistoricalleaders(team_id = '1611661328')
```

## Value

Return a named list of data frames: CareerLeadersByTeam

**CareerLeadersByTeam**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

# wnba_teaminfocommon Get WNBA Stats API Team Common Info

---

## Description

**Get WNBA Stats API Team Common Info**

**Get WNBA Stats API Team Common Info**

## Usage

```
wnba_teaminfocommon(
  league_id = "10",
  season = most_recent_wnba_season() - 1,
  season_type = "Regular Season",
  team_id = "1611661328",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '10'. Other options include '00': NBA, '20': G-League |
| season | Season - format 2020-21 |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | Team ID |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teaminfocommon(team_id = '1611661328')
```

## Value

Return a named list of data frames: AvailableSeasons, TeamInfoCommon, TeamSeasonRanks

**TeamInfoCommon**

See the HTML help or pkgdown reference for the column table.

**TeamSeasonRanks**

See the HTML help or pkgdown reference for the column table.

**AvailableSeasons**

| col_name | types | description |
|---|---|---|
| SEASON_ID | character | Unique season identifier. |

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teamplayeronoffdet wnba_teams(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

`wnba_teamplayeronoffdetails`

# Get WNBA Stats API Team Player On/Off Details

---

## Description

**Get WNBA Stats API Team Player On/Off Details**

**Get WNBA Stats API Team Player On/Off Summary**

**Get WNBA Stats API Team Player Dashboard**

**Get WNBA Stats API Team Player Dashboard**

## Usage

```
wnba_teamplayeronoffdetails(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
```

```
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamplayeronoffsummary(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)

wnba_teamplayerdashboard(
  date_from = "",
```

```
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  pace_adjust = "N",
  plus_minus = "N",
  po_round = "",
  per_mode = "Totals",
  period = 0,
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| pace_adjust | pace_adjust |
| plus_minus | plus_minus |
| po_round | po_round |
| per_mode | per_mode |
| period | period |
| rank | rank |

| | |
|---|---|
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamplayeronoffdetails(team_id = '1611661328')

wnba_teamplayeronoffsummary(team_id = '1611661328')

wnba_teamplayerdashboard(team_id = '1611661328')
```

## Value

Return a named list of data frames: OverallTeamPlayerOnOffDetails, PlayersOffCourtTeamPlayerOnOffDetails, PlayersOnCourtTeamPlayerOnOffDetails

**OverallTeamPlayerOnOffDetails**

Columns as documented in the shared wnba_team_summary_overall_schema table.

**PlayersOnCourtTeamPlayerOnOffDetails**

Columns as documented in the shared wnba_on_off_schema table.

**PlayersOffCourtTeamPlayerOnOffDetails**

Columns as documented in the shared wnba_on_off_schema table.

Return a named list of data frames: OverallTeamPlayerOnOffSummary, PlayersOffCourtTeamPlayerOnOffSummary, PlayersOnCourtTeamPlayerOnOffSummary

**OverallTeamPlayerOnOffSummary**

Columns as documented in the shared wnba_team_summary_overall_schema table.

**PlayersOnCourtTeamPlayerOnOffSummary**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| TEAM_ID | character | Unique team identifier. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| VS_PLAYER_ID | character | |
| VS_PLAYER_NAME | character | |
| COURT_STATUS | character | |
| GP | character | Games played. |

| | | |
|---|---|---|
| MIN | character | Minutes played. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| OFF_RATING | character | Offensive rating (points produced per 100 possessions). |
| DEF_RATING | character | Defensive rating (points allowed per 100 possessions). |
| NET_RATING | character | Net rating (off rating - def rating). |

**PlayersOffCourtTeamPlayerOnOffSummary**

Same columns as the **PlayersOnCourtTeamPlayerOnOffSummary** table above.

Return a named list of data frames: PlayersSeasonTotals, TeamOverall

**TeamOverall**

See the HTML help or pkgdown reference for the column table.

**PlayersSeasonTotals**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: `wnba_leaguedashteamstats()`, `wnba_teamdashboardbyclutch()`, `wnba_teamdashlineups()`, `wnba_teamdetails()`, `wnba_teamestimatedmetrics()`, `wnba_teamgamelog()`, `wnba_teamgamelogs()`, `wnba_teamgamestreakfinder()`, `wnba_teamhistoricalleaders()`, `wnba_teaminfocommon()`, `wnba_teams()`, `wnba_teamvsplayer()`, `wnba_teamyearbyyearstats()`

---

`wnba_teams` **Get WNBA Stats API Teams**

---

## Description

**Get WNBA Stats API Teams**

**Get WNBA Stats API Teams**

## Usage

```
wnba_teams(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teams()
```

## Value

Return a data frame with the following columns:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teamvsplayer(), wnba_teamyearbyyearstats()

---

`wnba_teamvsplayer` **Get WNBA Stats API Team vs Player**

---

## Description

**Get WNBA Stats API Team vs Player**

**Get WNBA Stats API Team vs Player**

## Usage

```
wnba_teamvsplayer(
  date_from = "",
  date_to = "",
  game_segment = "",
  last_n_games = 0,
  league_id = "10",
  location = "",
  measure_type = "Base",
  month = 0,
  opponent_team_id = 0,
  outcome = "",
  po_round = "",
  pace_adjust = "N",
  per_mode = "Totals",
  period = 0,
  player_id = "",
  plus_minus = "N",
  rank = "N",
  season = most_recent_wnba_season() - 1,
  season_segment = "",
  season_type = "Regular Season",
```

```
  shot_clock_range = "",
  team_id = "1611661328",
  vs_conference = "",
  vs_division = "",
  vs_player_id = "1628932",
  ...
)
```

## Arguments

| | |
|---|---|
| date_from | date_from |
| date_to | date_to |
| game_segment | game_segment |
| last_n_games | last_n_games |
| league_id | league_id |
| location | location |
| measure_type | measure_type |
| month | month |
| opponent_team_id | opponent_team_id |
| outcome | outcome |
| po_round | po_round |
| pace_adjust | pace_adjust |
| per_mode | per_mode |
| period | period |
| player_id | Player ID |
| plus_minus | plus_minus |
| rank | rank |
| season | season |
| season_segment | season_segment |
| season_type | season_type |
| shot_clock_range | shot_clock_range |
| team_id | team_id |
| vs_conference | vs_conference |
| vs_division | vs_division |
| vs_player_id | vs_player_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamvsplayer(team_id = '1611661328', vs_player_id = '1628932')
```

## Value

Return a named list of data frames: OnOffCourt, Overall, ShotAreaOffCourt, ShotAreaOnCourt, ShotAreaOverall, ShotDistanceOffCourt, ShotDistanceOnCourt, ShotDistanceOverall, vsPlayerOverall

### Overall

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| TEAM_ID | character | Unique team identifier. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| GP | character | Games played. |
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |
| FGM_RANK | character | |
| FGA_RANK | character | |
| FG_PCT_RANK | character | |
| FG3M_RANK | character | |
| FG3A_RANK | character | |
| FG3_PCT_RANK | character | |

| | |
|---|---|
| FTM_RANK | character |
| FTA_RANK | character |
| FT_PCT_RANK | character |
| OREB_RANK | character |
| DREB_RANK | character |
| REB_RANK | character |
| AST_RANK | character |
| TOV_RANK | character |
| STL_RANK | character |
| BLK_RANK | character |
| BLKA_RANK | character |
| PF_RANK | character |
| PFD_RANK | character |
| PTS_RANK | character |
| PLUS_MINUS_RANK | character |

**vsPlayerOverall**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| PLAYER_ID | character | Unique player identifier. |
| GP | character | Games played. |
| W | character | Wins. |
| L | character | Losses. |
| W_PCT | character | Wins percentage (0-1 decimal). |
| MIN | character | Minutes played. |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |
| FG3M | character | Three-point field goals made. |
| FG3A | character | Three-point field goal attempts. |
| FG3_PCT | character | Three-point field goal percentage (0-1). |
| FTM | character | Free throws made. |
| FTA | character | Free throw attempts. |
| FT_PCT | character | Free throw percentage (0-1). |
| OREB | character | Offensive rebounds. |
| DREB | character | Defensive rebounds. |
| REB | character | Total rebounds. |
| AST | character | Assists. |
| TOV | character | Turnovers. |
| STL | character | Steals. |
| BLK | character | Blocks. |
| BLKA | character | |
| PF | character | Personal fouls. |
| PFD | character | |
| PTS | character | Points scored. |

| | | |
|---|---|---|
| PLUS_MINUS | character | Plus/minus point differential while on court. |
| NBA_FANTASY_PTS | character | |
| DD2 | character | |
| TD3 | character | |
| WNBA_FANTASY_PTS | character | |
| GP_RANK | character | |
| W_RANK | character | |
| L_RANK | character | |
| W_PCT_RANK | character | |
| MIN_RANK | character | |
| FGM_RANK | character | |
| FGA_RANK | character | |
| FG_PCT_RANK | character | |
| FG3M_RANK | character | |
| FG3A_RANK | character | |
| FG3_PCT_RANK | character | |
| FTM_RANK | character | |
| FTA_RANK | character | |
| FT_PCT_RANK | character | |
| OREB_RANK | character | |
| DREB_RANK | character | |
| REB_RANK | character | |
| AST_RANK | character | |
| TOV_RANK | character | |
| STL_RANK | character | |
| BLK_RANK | character | |
| BLKA_RANK | character | |
| PF_RANK | character | |
| PFD_RANK | character | |
| PTS_RANK | character | |
| PLUS_MINUS_RANK | character | |
| NBA_FANTASY_PTS_RANK | character | |
| DD2_RANK | character | |
| TD3_RANK | character | |
| WNBA_FANTASY_PTS_RANK | character | |

**OnOffCourt**

Columns as documented in the shared wnba_on_off_schema table.

**ShotDistanceOverall**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| GROUP_VALUE | character | |
| TEAM_ID | character | Unique team identifier. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| FGM | character | Field goals made. |

| | | |
|---|---|---|
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |

**ShotDistanceOnCourt**

| col_name | types | description |
|---|---|---|
| GROUP_SET | character | |
| TEAM_ID | character | Unique team identifier. |
| TEAM_ABBREVIATION | character | Short team abbreviation (e.g. 'LAS'). |
| TEAM_NAME | character | Full team display name (e.g. 'Las Vegas Aces'). |
| VS_PLAYER_ID | character | |
| VS_PLAYER_NAME | character | |
| COURT_STATUS | character | |
| GROUP_VALUE | character | |
| FGM | character | Field goals made. |
| FGA | character | Field goal attempts. |
| FG_PCT | character | Field goal percentage (0-1). |

**ShotDistanceOffCourt**

Same columns as the **ShotDistanceOnCourt** table above.

**ShotAreaOverall**

Same columns as the **ShotDistanceOverall** table above.

**ShotAreaOnCourt**

Same columns as the **ShotDistanceOnCourt** table above.

**ShotAreaOffCourt**

Same columns as the **ShotDistanceOnCourt** table above.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: `wnba_leaguedashteamstats()`, `wnba_teamdashboardbyclutch()`, `wnba_teamdashlineups()`, `wnba_teamdetails()`, `wnba_teamestimatedmetrics()`, `wnba_teamgamelog()`, `wnba_teamgamelogs()`, `wnba_teamgamestreakfinder()`, `wnba_teamhistoricalleaders()`, `wnba_teaminfocommon()`, `wnba_teamplayeronoffdetails()`, `wnba_teams()`, `wnba_teamyearbyyearstats()`

wnba_teamyearbyyearstats

**Get WNBA Stats API Team Year by Year Stats**

## Description

**Get WNBA Stats API Team Year by Year Stats**

**Get WNBA Stats API Team Year by Year Stats**

## Usage

```
wnba_teamyearbyyearstats(
  league_id = "10",
  per_mode = "Totals",
  season_type = "Regular Season",
  team_id = "1611661328",
  ...
)
```

## Arguments

| | |
|---|---|
| league_id | League - default: '00'. Other options include '10': WNBA, '20': G-League |
| per_mode | Per Mode |
| season_type | Season Type - Regular Season, Playoffs, All-Star |
| team_id | Team ID |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_teamyearbyyearstats(team_id = '1611661328')
```

## Value

Return a named list of data frames: TeamStats

**TeamStats**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Team Functions: wnba_leaguedashteamstats(), wnba_teamdashboardbyclutch(), wnba_teamdashlineups(), wnba_teamdetails(), wnba_teamestimatedmetrics(), wnba_teamgamelog(), wnba_teamgamelogs(), wnba_teamgamestreakfinder(), wnba_teamhistoricalleaders(), wnba_teaminfocommon(), wnba_teamplayeronoffdetails(), wnba_teams(), wnba_teamvsplayer()

wnba_todays_scoreboard

**Get WNBA Stats API Today's Scoreboard**

## Description

**Get WNBA Stats API Today's Scoreboard**

**Get WNBA Stats API Today's Scoreboard**

## Usage

```
wnba_todays_scoreboard(...)
```

## Arguments

| | |
|---|---|
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_todays_scoreboard()
```

## Value

Return a data frame with the following columns:

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Schedule Functions: `wnba_schedule()`, `wnba_scoreboardv2()`, `wnba_scoreboardv3()`

Other WNBA Live Functions: `wnba_live_pbp()`

`wnba_videoevents` **Get WNBA Stats API Video Events**

## Description

**Get WNBA Stats API Video Events**

**Get WNBA Stats API Video Events**

## Usage

```
wnba_videoevents(game_id = "1022200075", game_event_id = "10", ...)
```

## Arguments

| | |
|---|---|
| `game_id` | game_id |
| `game_event_id` | game_event_id |
| `...` | Additional arguments passed to an underlying function like httr. |

## Details

(Possibly Defunct)

```
wnba_videoevents(game_id = '1022200075', game_event_id = '10')
```

## Value

Return a list of tibbles: videoUrls, playlist

## Author(s)

Saiem Gilani

## See Also

Other WNBA Video Functions: `wnba_videodetailsasset()`, `wnba_videostatus()`

wnba_videostatus **Get WNBA Stats API Video Status**

## Description

**Get WNBA Stats API Video Status**

**Get WNBA Stats API Video Status**

## Usage

```
wnba_videostatus(game_date = "2022-06-10", league_id = "10", ...)
```

## Arguments

| | |
|---|---|
| game_date | game_date |
| league_id | league_id |
| ... | Additional arguments passed to an underlying function like httr. |

## Details

```
wnba_videostatus(game_date = '2022-06-10', league_id = '10')
```

## Value

Return a list of tibbles: VideoStatus

**VideoStatus**

See the HTML help or pkgdown reference for the column table.

## Author(s)

Saiem Gilani

## See Also

Other WNBA Video Functions: wnba_videodetailsasset(), wnba_videoevents()

year_to_season **Convert a calendar year to a WNBA / NBA season string**

## Description

Returns a season string of the form YYYY-YY (e.g. 2024 -> "2024-25"). WNBA seasons span a single calendar year, but several Stats API endpoints (and several NBA-derived endpoints used in load helpers) accept the two-year season-string form, so this helper is provided for parity with the analogous helper in the hoopR package.

## Usage

```
year_to_season(year)
```

## Arguments

| | |
|---|---|
| year | a four-digit calendar year (numeric or character). |

## Value

A character season string, e.g. "2024-25".

# Index


∗ **API**
wnba_stats, 873
∗ **Atrium Functions**
atrium_box_score, 12
atrium_competition_player_stats, 14
atrium_competition_team_stats, 16
atrium_end_year, 18
atrium_fixture, 19
atrium_fixture_periods, 21
atrium_fixtures, 23
atrium_fixtures_ribbon, 26
atrium_leaders, 28
atrium_pbp, 30
atrium_player, 32
atrium_player_game_log, 34
atrium_player_seasons, 37
atrium_player_stats, 39
atrium_players, 41
atrium_season_id, 43
atrium_seasons, 44
atrium_shots, 46
atrium_standings, 48
atrium_standings_summary, 50
atrium_team, 52
atrium_team_box_score, 55
atrium_team_stats, 57
atrium_teams, 59
lmb_competition, 451
lnbp_competition, 457
lpb_competition, 497
nbl1_competition, 560
∗ **Bart Torvik Functions**
bart_wbb_game_schedule, 61
bart_wbb_ratings, 62
∗ **Box**
espn_wbb_game_all, 81
espn_wnba_game_all, 171
∗ **Conference and Division Group loader functions**
load_wbb_group_aliases, 470
load_wbb_group_seasons, 471
load_wbb_groups, 472
load_wbb_team_group_seasons, 478
∗ **Conferences**
espn_wbb_conferences, 77
espn_wnba_conferences, 160
∗ **ESPN Basketball**
espn_basketball, 66
espn_basketball_player_core, 67
∗ **ESPN WBB Functions**
espn_wbb_athletes_index, 69
espn_wbb_award, 70
espn_wbb_calendar, 71
espn_wbb_coach, 73
espn_wbb_coach_record, 74
espn_wbb_coach_season, 75
espn_wbb_coaches, 76
espn_wbb_conferences, 77
espn_wbb_franchise, 78
espn_wbb_franchises, 80
espn_wbb_game_all, 81
espn_wbb_game_odds, 83
espn_wbb_game_official_detail, 85
espn_wbb_game_play, 86
espn_wbb_game_play_personnel, 87
espn_wbb_game_player_box, 88
espn_wbb_game_probabilities, 90
espn_wbb_game_propbets, 91
espn_wbb_game_team_linescores, 92
espn_wbb_game_team_roster_entry, 94
espn_wbb_injuries, 95
espn_wbb_leaders, 97
espn_wbb_news, 98
espn_wbb_player_awards, 99
espn_wbb_player_career_stats, 100
espn_wbb_player_eventlog_v2, 101

espn_wbb_player_info, 103
espn_wbb_player_overview, 105
espn_wbb_player_seasons, 108
espn_wbb_player_stats, 109
espn_wbb_position, 110
espn_wbb_positions, 111
espn_wbb_powerindex, 112
espn_wbb_rankings, 113
espn_wbb_scoreboard, 114
espn_wbb_season_awards, 115
espn_wbb_season_group, 117
espn_wbb_season_groups, 119
espn_wbb_season_info, 120
espn_wbb_season_leaders, 122
espn_wbb_season_ranking, 123
espn_wbb_season_rankings, 124
espn_wbb_season_type, 126
espn_wbb_season_types, 127
espn_wbb_season_week, 128
espn_wbb_season_weeks, 129
espn_wbb_seasons, 131
espn_wbb_standings, 132
espn_wbb_team, 133
espn_wbb_team_injuries, 135
espn_wbb_team_news, 136
espn_wbb_team_record_detail, 138
espn_wbb_team_schedule, 139
espn_wbb_team_season_roster, 140
espn_wbb_team_season_statistics, 142
espn_wbb_team_stats, 143
espn_wbb_teams, 144
espn_wbb_tournament, 145
espn_wbb_tournament_season, 146
espn_wbb_tournament_seasons, 147
espn_wbb_tournaments, 149
espn_wbb_venues, 150
espn_wbb_week_ranking, 151
espn_wbb_week_rankings, 152

∗ **ESPN WNBA Functions**

espn_wnba_athletes_index, 154
espn_wnba_award, 155
espn_wnba_calendar, 156
espn_wnba_coach_season, 157
espn_wnba_coaches, 159
espn_wnba_conferences, 160
espn_wnba_draft, 161
espn_wnba_draft_athlete_detail, 162
espn_wnba_draft_pick, 164
espn_wnba_draft_rounds, 165
espn_wnba_franchise, 167
espn_wnba_franchises, 168
espn_wnba_freeagents, 169
espn_wnba_futures, 170
espn_wnba_game_all, 171
espn_wnba_game_odds, 173
espn_wnba_game_official_detail, 175
espn_wnba_game_play, 176
espn_wnba_game_play_personnel, 178
espn_wnba_game_player_box, 179
espn_wnba_game_probabilities, 180
espn_wnba_game_propbets, 181
espn_wnba_game_team_linescores, 183
espn_wnba_game_team_roster_entry, 185
espn_wnba_injuries, 186
espn_wnba_leaders, 187
espn_wnba_news, 188
espn_wnba_player_awards, 190
espn_wnba_player_career_stats, 191
espn_wnba_player_eventlog_v2, 192
espn_wnba_player_info, 193
espn_wnba_player_overview, 195
espn_wnba_player_seasons, 198
espn_wnba_player_stats, 200
espn_wnba_position, 201
espn_wnba_positions, 202
espn_wnba_powerindex, 203
espn_wnba_scoreboard, 204
espn_wnba_season_awards, 205
espn_wnba_season_group, 207
espn_wnba_season_groups, 209
espn_wnba_season_info, 210
espn_wnba_season_leaders, 212
espn_wnba_season_ranking, 213
espn_wnba_season_rankings, 215
espn_wnba_season_type, 216
espn_wnba_season_types, 217
espn_wnba_season_week, 218
espn_wnba_season_weeks, 220
espn_wnba_seasons, 221
espn_wnba_standings, 222
espn_wnba_team, 224

espn_wnba_team_injuries, 226
espn_wnba_team_news, 227
espn_wnba_team_record, 228
espn_wnba_team_record_detail, 230
espn_wnba_team_schedule, 231
espn_wnba_team_season_roster, 232
espn_wnba_team_season_statistics, 234
espn_wnba_team_stats, 235
espn_wnba_teams, 236
espn_wnba_transactions, 237
espn_wnba_venues, 238
espn_wnba_week_ranking, 240
espn_wnba_week_rankings, 241
* **ESPN**
espn_basketball, 66
* **FEB Functions**
feb_game_info, 243
feb_leaders, 245
feb_overview, 246
feb_pbp, 248
feb_period_scores, 250
feb_player_box, 251
feb_results, 253
feb_schedule, 255
feb_shots, 257
feb_standings, 259
feb_team_stats, 261
* **FIBA Competition Functions**
fiba_ecw_competition, 263
fiba_elw_competition, 271
fiba_wafrobasket_competition, 361
fiba_wamericup_competition, 370
fiba_wasiacup_competition, 378
fiba_weurobasket_competition, 387
fiba_woly_competition, 395
fiba_wwc_competition, 404
* **FIBA Functions**
fiba_gamepage_box_players, 280
fiba_gamepage_box_teams, 283
fiba_gamepage_lineup, 286
fiba_gamepage_pbp, 288
fiba_gamepage_period_scores, 290
fiba_gamepage_shots, 292
fiba_gdap_awards, 294
fiba_gdap_competition, 296
fiba_gdap_editions, 298
fiba_gdap_federation_games, 301
fiba_gdap_federation_ranking, 304
fiba_gdap_final_standings, 306
fiba_gdap_game, 307
fiba_gdap_game_officials, 310
fiba_gdap_game_periods, 311
fiba_gdap_games_between, 313
fiba_gdap_gateway_competition_ids, 316
fiba_gdap_gateway_federation_ranking, 318
fiba_gdap_medal_count, 320
fiba_gdap_national_teams, 321
fiba_gdap_nominations, 323
fiba_gdap_player_game_stats, 325
fiba_gdap_player_info, 327
fiba_gdap_player_stats, 329
fiba_gdap_ranking_zones, 332
fiba_gdap_roster, 333
fiba_gdap_roster_coaches, 335
fiba_gdap_roster_latest, 337
fiba_gdap_round_gamedays, 339
fiba_gdap_round_standings, 341
fiba_gdap_rounds, 343
fiba_gdap_schedule, 345
fiba_gdap_team_games, 348
fiba_gdap_team_player_stats, 351
fiba_gdap_team_staff, 354
fiba_gdap_team_stats, 356
fiba_gdap_teams, 359
* **Fox Sports Functions**
fox_basketball_league_leaders, 412
fox_wbb_teams_all, 413
fox_wnba_pbp, 414
fox_wnba_team_roster, 415
* **Functions**
wnba_possession_lineups, 856
wnba_rapm, 856
wnba_shot_zones, 864
* **Game**
espn_wbb_game_all, 81
espn_wnba_game_all, 171
* **Genius Functions**
genius_competition_games, 417
genius_competitions, 419
genius_game, 421
genius_game_crosswalk, 422
genius_match_competition, 424
genius_pbp, 425

genius_player_box, 427
genius_schedule, 430
genius_schedule_embed, 432
genius_shots, 433
genius_team_box, 435
genius_team_crosswalk, 438
genius_widget_games, 439
lfb_competition, 445
wbbl_competition, 684
* **Her Hoop Stats Functions**
hhs_team_roster, 442
hhs_team_stats, 443
hhs_teams, 444
* **Internal**
check_status, 66
csv_from_url, 66
parquet_from_url, 570
progressively, 570
rds_from_url, 571
year_to_season, 922
* **Lineup**
wnba_possession_lineups, 856
wnba_rapm, 856
wnba_shot_zones, 864
* **Loaders**
wehoop_loaders, 690
* **Model Dataset loader functions**
load_wnba_player_impact, 485
* **NBA Boxscore Functions**
wnba_live_boxscore, 808
* **NBA Live Functions**
wnba_live_boxscore, 808
* **NBL Rosetta Functions**
nbl_rosetta_match, 505
nbl_rosetta_match_by_id, 508
nbl_rosetta_match_pbp, 510
nbl_rosetta_match_player_box, 512
nbl_rosetta_match_player_period_box, 515
nbl_rosetta_match_team_box, 517
nbl_rosetta_matches, 519
nbl_rosetta_next_match, 523
nbl_rosetta_next_match_team, 525
nbl_rosetta_next_matches, 528
nbl_rosetta_outcomes, 530
nbl_rosetta_player_game_log, 532
nbl_rosetta_player_stats, 535
nbl_rosetta_players, 539
nbl_rosetta_seasons, 541
nbl_rosetta_seasons_current, 543
nbl_rosetta_standings, 545
nbl_rosetta_stats_leaders, 547
nbl_rosetta_team_players, 550
nbl_rosetta_team_stats, 552
nbl_rosetta_team_stats_leaders, 555
nbl_rosetta_teams, 558
* **NCAA WBB Functions**
ncaa_wbb_NET_rankings, 568
* **NCAA WBB loader functions**
load_ncaa_wbb_pbp, 464
* **NCAA WBB**
ncaa_wbb, 567
* **NCAA**
ncaa_wbb, 567
ncaa_wbb_NET_rankings, 568
* **NET**
ncaa_wbb_NET_rankings, 568
* **PBP**
espn_wbb_game_all, 81
espn_wnba_game_all, 171
* **Player**
espn_wbb_game_all, 81
espn_wbb_player_stats, 109
espn_wnba_game_all, 171
espn_wnba_player_stats, 200
* **Rankings**
espn_wbb_rankings, 113
ncaa_wbb_NET_rankings, 568
* **Roster**
espn_wbb_game_all, 81
espn_wnba_game_all, 171
* **SCB Functions**
scb_game, 571
scb_game_actions, 575
scb_game_players, 577
scb_game_teams, 579
scb_games, 582
scb_games_recent, 586
scb_player_statistics, 589
scb_playoff_games, 592
scb_playoff_series, 594
scb_records_per_game, 596
scb_records_season_averages, 598
scb_referees, 599
scb_season_player, 601

scb_season_players, 603
scb_season_teams, 604
scb_seasons, 606
scb_standings, 608
scb_team, 610
scb_team_rosters, 612
scb_team_seasons, 614
scb_team_statistics, 615
scb_transfers, 618
scb_transfers_teams, 620
* **Scoreboard**
espn_wbb_scoreboard, 114
espn_wnba_scoreboard, 204
* **Standings**
espn_wbb_standings, 132
espn_wnba_standings, 222
* **Stats**
espn_wbb_player_stats, 109
espn_wbb_team_stats, 143
espn_wnba_player_stats, 200
espn_wnba_team_stats, 235
wnba_stats, 873
* **TBF Functions**
tbf_daily_matches, 621
tbf_game_header, 623
tbf_game_leaders, 625
tbf_game_players, 627
tbf_highlights, 628
tbf_leaders_assists, 630
tbf_leaders_blocks, 632
tbf_leaders_efficiency, 633
tbf_leaders_minutes, 635
tbf_leaders_points, 637
tbf_leaders_rebounds, 638
tbf_leaders_steals, 640
tbf_leaders_threes, 642
tbf_leaders_turnovers, 643
tbf_league_seasons, 645
tbf_leagues, 647
tbf_pbp, 648
tbf_period_scores, 650
tbf_player_box, 651
tbf_player_career, 654
tbf_players, 656
tbf_schedule, 657
tbf_seasons, 659
tbf_shots, 661
tbf_standings, 662
tbf_team_box, 664
tbf_team_games, 666
tbf_team_player_stats, 668
tbf_team_roster, 671
tbf_team_staff, 673
tbf_teams, 674
tbf_weeks, 676
* **Teams**
espn_wbb_teams, 144
espn_wnba_teams, 236
* **Team**
espn_wbb_game_all, 81
espn_wbb_team_stats, 143
espn_wnba_game_all, 171
espn_wnba_team_stats, 235
* **WBB Crosswalk Functions**
load_wnba_team_crosswalk, 495
wbb_player_crosswalk, 682
wbb_schedule_crosswalk, 682
wbb_team_crosswalk, 683
* **WBB loader functions**
load_wbb_pbp, 473
* **WBB**
espn_wbb_conferences, 77
espn_wbb_game_all, 81
espn_wbb_player_stats, 109
espn_wbb_rankings, 113
espn_wbb_scoreboard, 114
espn_wbb_standings, 132
espn_wbb_team_stats, 143
espn_wbb_teams, 144
ncaa_wbb, 567
ncaa_wbb_NET_rankings, 568
* **WJBL Functions**
wjbl_box_score, 691
wjbl_game, 693
wjbl_game_info, 695
wjbl_game_months, 698
wjbl_head_to_head, 700
wjbl_leagues, 702
wjbl_pbp, 704
wjbl_period_scores, 706
wjbl_player, 708
wjbl_player_period_leaders, 710
wjbl_roster, 712
wjbl_schedule, 714
wjbl_staff, 717
wjbl_standing_seasons, 719

wjbl_team_box_score, 721
wjbl_team_period_stats, 723
wjbl_team_ranking, 725
wjbl_team_standings, 728
wjbl_teams, 730
* **WNBA Basketball-Reference Functions**
bref_wnba_player_stats, 63
bref_wnba_standings, 64
bref_wnba_team_stats, 65
* **WNBA Boxscore Functions**
wnba_boxscoretraditionalv2, 740
wnba_gamerotation, 768
* **WNBA Boxscore V3 Functions**
wnba_boxscoresummaryv3, 739
wnba_boxscoretraditionalv3, 743
* **WNBA Clutch Functions**
wnba_leaguedashteamclutch, 786
wnba_playerdashboardbyclutch, 821
wnba_teamdashboardbyclutch, 875
* **WNBA Crosswalk Functions**
load_wnba_team_crosswalk, 495
wnba_player_crosswalk, 813
wnba_schedule_crosswalk, 860
wnba_team_crosswalk, 874
* **WNBA Cume Functions**
wnba_cumestatsplayer, 752
wnba_cumestatsplayergames, 753
wnba_cumestatsteam, 754
wnba_cumestatsteamgames, 755
* **WNBA Draft Functions**
wnba_draftboard, 758
wnba_draftcombinestats, 759
wnba_drafthistory, 761
* **WNBA Fantasy Functions**
wnba_fantasywidget, 762
wnba_infographicfanduelplayer, 771
wnba_playerfantasyprofile, 835
wnba_playerfantasyprofilebargraph, 837
* **WNBA Franchise Functions**
wnba_franchisehistory, 764
wnba_franchiseleaders, 765
wnba_franchiseleaderswrank, 766
wnba_franchiseplayers, 767
* **WNBA Game Finder Functions**
wnba_leaguegamefinder, 793
wnba_playergamestreakfinder, 841
wnba_teamgamestreakfinder, 895
* **WNBA Leaders Functions**
wnba_alltimeleadersgrids, 731
wnba_assistleaders, 736
wnba_assisttracker, 738
wnba_homepagewidget, 769
wnba_leagueleaders, 800
* **WNBA League Functions**
wnba_leaguedashlineups, 772
wnba_leaguedashplayerbiostats, 774
wnba_leaguedashplayerclutch, 777
wnba_leaguedashplayershotlocations, 780
wnba_leaguedashplayerstats, 783
wnba_leaguedashteamclutch, 786
wnba_leaguedashteamshotlocations, 788
wnba_leaguedashteamstats, 791
wnba_leaguegamefinder, 793
wnba_leaguegamelog, 798
wnba_leaguelineupviz, 801
wnba_leagueplayerondetails, 803
wnba_leagueseasonmatchups, 805
wnba_leaguestandingsv3, 806
wnba_shotchartleaguewide, 868
* **WNBA Lineup Functions**
wnba_gamerotation, 768
wnba_leaguedashlineups, 772
wnba_leaguelineupviz, 801
wnba_possession_lineups, 856
wnba_rapm, 856
wnba_shot_zones, 864
wnba_shotchartlineupdetail, 869
wnba_teamdashlineups, 886
* **WNBA Live Functions**
wnba_live_pbp, 809
wnba_todays_scoreboard, 919
* **WNBA Officiating Functions**
wnba_referee_assignments, 857
* **WNBA PBP Functions**
wnba_data_pbp, 757
wnba_live_pbp, 809
wnba_pbp, 810
wnba_pbps, 811
wnba_playbyplayv3, 812
* **WNBA Player Functions**
wnba_infographicfanduelplayer, 771
wnba_leaguedashplayerbiostats, 774
wnba_leaguedashplayerclutch, 777

wnba_leaguedashplayershotlocations, 780
wnba_leaguedashplayerstats, 783
wnba_leagueplayerondetails, 803
wnba_leagueseasonmatchups, 805
wnba_playerawards, 813
wnba_playercareerbycollege, 814
wnba_playercareerbycollegerollup, 816
wnba_playercareerstats, 817
wnba_playercompare, 819
wnba_playerdashboardbyclutch, 821
wnba_playerestimatedmetrics, 834
wnba_playerfantasyprofile, 835
wnba_playerfantasyprofilebargraph, 837
wnba_playergamelog, 838
wnba_playergamelogs, 839
wnba_playergamestreakfinder, 841
wnba_playerheadshot, 846
wnba_playerindex, 847
wnba_playernextngames, 849
wnba_playerprofilev2, 850
wnba_playervsplayer, 853
* **WNBA Roster Functions**
wnba_commonallplayers, 748
wnba_commonplayerinfo, 749
wnba_commonteamroster, 751
* **WNBA Schedule Functions**
wnba_schedule, 859
wnba_scoreboardv2, 860
wnba_scoreboardv3, 863
wnba_todays_scoreboard, 919
* **WNBA Shooting Functions**
wnba_leaguedashplayershotlocations, 780
wnba_leaguedashteamshotlocations, 788
wnba_playerdashboardbyclutch, 821
wnba_shotchartdetail, 866
wnba_shotchartleaguewide, 868
wnba_shotchartlineupdetail, 869
wnba_teamdashboardbyclutch, 875
* **WNBA Stats loader functions**
load_wnba_stats_leaguedash, 487
load_wnba_stats_rosters, 488
* **WNBA Stats**
wnba_stats, 873
* **WNBA Team Functions**
wnba_leaguedashteamstats, 791
wnba_teamdashboardbyclutch, 875
wnba_teamdashlineups, 886
wnba_teamdetails, 889
wnba_teamestimatedmetrics, 891
wnba_teamgamelog, 892
wnba_teamgamelogs, 893
wnba_teamgamestreakfinder, 895
wnba_teamhistoricalleaders, 905
wnba_teaminfocommon, 906
wnba_teamplayeronoffdetails, 907
wnba_teams, 911
wnba_teamvsplayer, 912
wnba_teamyearbyyearstats, 918
* **WNBA Video Functions**
wnba_videoevents, 920
wnba_videostatus, 921
* **WNBA loader functions**
load_wnba_pbp, 480
* **WNBA**
espn_wnba_conferences, 160
espn_wnba_game_all, 171
espn_wnba_player_stats, 200
espn_wnba_scoreboard, 204
espn_wnba_standings, 222
espn_wnba_team_stats, 235
espn_wnba_teams, 236
wnba_possession_lineups, 856
wnba_rapm, 856
wnba_shot_zones, 864
wnba_stats, 873
* **datasets**
parameter_descriptions, 569
* **wehoop Loaders**
wehoop_loaders, 690
.players_on_court, 12

arrow::read_parquet(), *570*
atrium_box_score, 12
atrium_box_score(), *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *454*, *457*, *460*, *464*, *500*, *503*, *563*, *567*
atrium_competition_player_stats, 14
atrium_competition_player_stats(), *14*, *17*, *18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*

atrium_competition_team_stats, 16
atrium_competition_team_stats(), *14*, *16*, *18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_end_year, 18
atrium_end_year(), *12*, *14–17*, *19*, *21–25*, *27–36*, *38–47*, *49–53*, *55–61*, *457*, *464*, *503*, *567*
atrium_fixture, 19
atrium_fixture(), *14*, *16–18*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_fixture_periods, 21
atrium_fixture_periods(), *14*, *16–18*, *21*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_fixtures, 23
atrium_fixtures(), *12*, *14*, *16–19*, *21*, *23*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–47*, *50*, *52*, *55*, *57*, *59*, *61*, *451*, *452*, *457–459*, *464*, *498*, *503*, *561*, *562*, *567*
atrium_fixtures_ribbon, 26
atrium_fixtures_ribbon(), *14*, *16–18*, *21*, *23*, *25*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_leaders, 28
atrium_leaders(), *14*, *16–18*, *21*, *23*, *25*, *28*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_pbp, 30
atrium_pbp(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *34*, *36*, *39*, *41*, *43–47*, *50*, *52*, *55*, *57*, *59*, *61*, *456*, *457*, *463*, *464*, *502*, *503*, *566*, *567*
atrium_player, 32
atrium_player(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_player_game_log, 34
atrium_player_game_log(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_player_seasons, 37
atrium_player_seasons(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *503*, *567*
atrium_player_stats, 39
atrium_player_stats(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *504*, *567*
atrium_players, 41
atrium_players(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34–37*, *39*, *41*, *44*, *45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *504*, *567*
atrium_season_id, 43
atrium_season_id(), *12*, *14–19*, *21–25*, *27–36*, *38–43*, *45–47*, *49–53*, *55–61*, *457*, *464*, *504*, *567*
atrium_seasons, 44
atrium_seasons(), *12*, *14–19*, *21–53*, *55–61*, *452*, *457*, *458*, *464*, *498*, *504*, *561*, *567*
atrium_shots, 46
atrium_shots(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *50*, *52*, *55*, *57*, *59*, *61*, *456*, *457*, *463*, *464*, *503*, *504*, *566*, *567*
atrium_standings, 48
atrium_standings(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *453*, *457*, *460*, *464*, *499*, *504*, *563*, *567*
atrium_standings_summary, 50
atrium_standings_summary(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *55*, *57*, *59*, *61*, *457*, *464*, *504*, *567*
atrium_team, 52
atrium_team(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *57*, *59*, *61*, *457*, *464*, *504*, *567*
atrium_team_box_score, 55
atrium_team_box_score(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *59*, *61*, *455*, *457*, *461*, *464*, *501*, *504*, *564*, *567*
atrium_team_stats, 57
atrium_team_stats(), *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *36*, *39*, *41*, *43–45*, *47*,

*50*, *52*, *55*, *57*, *61*, *457*, *464*, *504*, *567*
atrium_teams, 59
atrium_teams(), *14*, *16–18*, *21*, *23–26*, *28–30*, *32*, *34*, *37*, *39*, *41–45*, *47*, *50*, *52, 53*, *55*, *57*, *59*, *454*, *457*, *460*, *464*, *500*, *504*, *563*, *567*

bart_wbb_game_schedule, 61
bart_wbb_game_schedule(), *62*
bart_wbb_ratings, 62
bart_wbb_ratings(), *62*
basketball_load_wbb_game_rosters_schema, *477*, *484*
basketball_load_wbb_officials_schema, *477*, *484*
basketball_load_wbb_shots_schema, *477*, *484*
basketball_load_wbb_standings_schema, *477*, *484*
basketball_load_wbb_team_stats_schema, *477*, *484*
bref_wnba_player_stats, 63
bref_wnba_player_stats(), *64, 65*
bref_wnba_standings, 64
bref_wnba_standings(), *63*, *65*
bref_wnba_team_stats, 65
bref_wnba_team_stats(), *63, 64*

check_status, 66
csv_from_url, 66

DBI::dbConnect(), *468*, *470–472*, *477*, *479*, *484*, *486*, *488*, *493*, *678*, *680*

espn_basketball, 66, *68*
espn_basketball_athletes_index_schema, *70*, *154*
espn_basketball_calendar_schema, *72*, *157*
espn_basketball_coach_season_schema, *75*, *158*
espn_basketball_conferences_schema, *78*, *160*
espn_basketball_game_all_player_player_schema, *82*
espn_basketball_game_all_player_schema, *172*
espn_basketball_game_all_plays_plays_schema, *81, 82*, *441*
espn_basketball_game_all_plays_schema, *172*, *441*
espn_basketball_game_broadcasts_schema, *84*, *174*
espn_basketball_game_odds_schema, *84*, *174*
espn_basketball_game_officials_schema, *84*, *174*
espn_basketball_game_probabilities_schema, *90*, *181*
espn_basketball_injuries_schema, *96*, *186*
espn_basketball_leaders_schema, *97*, *188*
espn_basketball_news_schema, *98*, *137*, *189*, *228*
espn_basketball_player_awards_schema, *99*, *190*
espn_basketball_player_career_stats_schema, *101*, *191*
espn_basketball_player_core, 67
espn_basketball_player_core(), *67*
espn_basketball_player_info_bio_schema, *103*, *194*
espn_basketball_player_overview_next_game_schema, *106*, *134*, *196*, *224*
espn_basketball_player_overview_statistics_schema, *105, 106*, *196, 197*
espn_basketball_scoreboard_schema, *115*, *205*
espn_basketball_season_group_schema, *118*, *208*
espn_basketball_season_leaders_schema, *122*, *213*
espn_basketball_season_ranking_schema, *124*, *214*
espn_basketball_season_type_schema, *126*, *216*
espn_basketball_season_week_schema, *129*, *219*
espn_basketball_team_box_schema, *81, 82*, *172*
espn_basketball_team_info_schema, *133*, *224*
espn_basketball_team_injuries_injuries_schema, *136*, *226*
espn_basketball_team_leaders_schema, *134*, *225*
espn_basketball_team_roster_schema,

*134*, *225*
espn_basketball_team_schedule_schema, *139*, *231*
espn_basketball_venues_schema, *150*, *239*
espn_basketball_week_ranking_schema, *151*, *240*
espn_wbb *(*espn_basketball*)*, 66
espn_wbb_athletes_index, 69
espn_wbb_athletes_index(), *71–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_award, 70
espn_wbb_award(), *70*, *72–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_calendar, 71
espn_wbb_calendar(), *70, 71*, *73, 74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_coach, 73
espn_wbb_coach(), *70–72*, *74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_coach_record, 74
espn_wbb_coach_record(), *70–73*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_coach_season, 75
espn_wbb_coach_season(), *70–74*, *77–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_coaches, 76
espn_wbb_coaches(), *70–74*, *76*, *78–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_conferences, 77
espn_wbb_conferences(), *67*, *70–74*, *76, 77*, *79, 80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_franchise, 78
espn_wbb_franchise(), *70–74*, *76–78*, *80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_franchises, 80
espn_wbb_franchises(), *70–74*, *76–79*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_all, 81
espn_wbb_game_all(), *67*, *70–74*, *76–80*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_broadcasts *(*espn_wbb_game_odds*)*, 83
espn_wbb_game_odds, 83
espn_wbb_game_odds(), *70–74*, *76–80*, *82*, *85*, *87–93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_official_detail, 85
espn_wbb_game_official_detail(), *70–74*, *76–80*, *82*, *84*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*,

*118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_officials (espn_wbb_game_odds), 83
espn_wbb_game_officials(), *85*
espn_wbb_game_play, 86
espn_wbb_game_play(), *70–74*, *76–80*, *82*, *84, 85*, *88–90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_play_personnel, 87
espn_wbb_game_play_personnel(), *70–74*, *76–80*, *82*, *84, 85*, *87*, *89, 90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_player_box, 88
espn_wbb_game_player_box(), *70–74*, *76–80*, *82*, *84, 85*, *87, 88*, *90*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_powerindex (espn_wbb_game_odds), 83
espn_wbb_game_predictor (espn_wbb_game_odds), 83
espn_wbb_game_probabilities, 90
espn_wbb_game_probabilities(), *70–74*, *76–80*, *82*, *84, 85*, *87–89*, *92, 93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_propbets, 91
espn_wbb_game_propbets(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *93*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_rosters (espn_wbb_game_all), 81
espn_wbb_game_rosters(), *67*
espn_wbb_game_situation (espn_wbb_game_odds), 83
espn_wbb_game_team_leaders (espn_wbb_game_team_linescores), 92
espn_wbb_game_team_linescores, 92
espn_wbb_game_team_linescores(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92*, *95–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_team_records (espn_wbb_game_team_linescores), 92
espn_wbb_game_team_roster (espn_wbb_game_team_linescores), 92
espn_wbb_game_team_roster(), *94*
espn_wbb_game_team_roster_entry, 94
espn_wbb_game_team_roster_entry(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *96–98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_game_team_score (espn_wbb_game_team_linescores), 92
espn_wbb_game_team_statistics (espn_wbb_game_team_linescores), 92
espn_wbb_game_team_statistics(), *88*
espn_wbb_injuries, 95
espn_wbb_injuries(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95*, *97, 98*, *100–102*, *104*, *107–109*, *111–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_leaders, 97
espn_wbb_leaders(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95, 96*, *98*, *100–102*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_news, 98

espn_wbb_news(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–97*, *100–102*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_pbp *(*espn_wbb_game_all*)*, 81
espn_wbb_pbp(), *67*, *86*
espn_wbb_player_awards, 99
espn_wbb_player_awards(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *101, 102*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_box *(*espn_wbb_game_all*)*, 81
espn_wbb_player_box(), *67*
espn_wbb_player_career_stats, 100
espn_wbb_player_career_stats(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100*, *102*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–132*, *134*, *136–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_eventlog *(*espn_wbb_player_overview*)*, 105
espn_wbb_player_eventlog(), *102*
espn_wbb_player_eventlog_v2, 101
espn_wbb_player_eventlog_v2(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100, 101*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_gamelog *(*espn_wbb_player_overview*)*, 105
espn_wbb_player_info, 103
espn_wbb_player_info(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_overview, 105
espn_wbb_player_overview(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *108*, *110–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_seasons, 108
espn_wbb_player_seasons(), *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107*, *110–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_splits *(*espn_wbb_player_overview*)*, 105
espn_wbb_player_statisticslog *(*espn_wbb_player_overview*)*, 105
espn_wbb_player_stats, 109
espn_wbb_player_stats(), *67*, *70–74*, *76–80*, *82*, *84, 85*, *87–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *111–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_player_stats_v3 *(*espn_wbb_player_overview*)*, 105
espn_wbb_position, 110
espn_wbb_position(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_positions, 111
espn_wbb_positions(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110, 111*, *113–116*, *118*, *120, 121*, *123–126*, *128–130*, *132*, *135–138*, *140–142*, *144–150*, *152, 153*
espn_wbb_powerindex, 112
espn_wbb_powerindex(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–112*, *114–116*, *118*, *120*, *122–125*, *127–130*, *132*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_rankings, 113
espn_wbb_rankings(), *67*, *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–113*, *115, 116*, *118*, *120*, *122–125*, *127–130*, *132*, *135–138*, *140, 141*, *143–149*,

*151–153*
espn_wbb_scoreboard, 114
espn_wbb_scoreboard(), *67*, *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–114*, *116*, *118*, *120*, *122–125*, *127–130*, *132*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_awards, 115
espn_wbb_season_awards(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–115*, *118*, *120*, *122–125*, *127–130*, *132*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_group, 117
espn_wbb_season_group(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–116*, *120*, *122–125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_group_children *(*espn_wbb_season_group*)*, 117
espn_wbb_season_group_teams *(*espn_wbb_season_group*)*, 117
espn_wbb_season_groups, 119
espn_wbb_season_groups(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–98*, *100–102*, *104*, *107, 108*, *110–116*, *119*, *122–125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_info, 120
espn_wbb_season_info(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *123–125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_leaders, 122
espn_wbb_season_leaders(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122*, *124, 125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_ranking, 123
espn_wbb_season_ranking(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_rankings, 124
espn_wbb_season_rankings(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122–124*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_type, 126
espn_wbb_season_type(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_types, 127
espn_wbb_season_types(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122–125*, *127*, *129, 130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_week, 128
espn_wbb_season_week(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107, 108*, *110–116*, *119, 120*, *122–125*, *127, 128*, *130*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_season_weeks, 129
espn_wbb_season_weeks(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–129*, *132, 133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_seasons, 131
espn_wbb_seasons(), *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *133*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_standings, 132

espn_wbb_standings(), *67*, *70–74*, *76–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132*, *135–138*, *140, 141*, *143–149*, *151–153*
espn_wbb_team, 133
espn_wbb_team(), *70–73*, *75–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *136–138*, *140, 141*, *143–148*, *150–153*
espn_wbb_team_box *(*espn_wbb_game_all*)*, 81
espn_wbb_team_box(), *67*
espn_wbb_team_injuries, 135
espn_wbb_team_injuries(), *70–73*, *75–80*, *82*, *84*, *86–90*, *92, 93*, *95–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135*, *137, 138*, *140, 141*, *143–148*, *150–153*
espn_wbb_team_leaders *(*espn_wbb_team*)*, 133
espn_wbb_team_news, 136
espn_wbb_team_news(), *70–73*, *75–80*, *82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135, 136*, *138*, *140, 141*, *143–148*, *150–153*
espn_wbb_team_record_detail, 138
espn_wbb_team_record_detail(), *70–73*, *75–80*, *82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *140, 141*, *143–148*, *150–153*
espn_wbb_team_roster *(*espn_wbb_team*)*, 133
espn_wbb_team_roster(), *140*
espn_wbb_team_schedule, 139
espn_wbb_team_schedule(), *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–138*, *141*, *143–148*, *150–153*
espn_wbb_team_season_profile *(*espn_wbb_team*)*, 133
espn_wbb_team_season_roster, 140
espn_wbb_team_season_roster(), *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139, 140*, *143–148*, *150–153*
espn_wbb_team_season_statistics, 142
espn_wbb_team_season_statistics(), *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *144–148*, *150–153*
espn_wbb_team_stats, 143
espn_wbb_team_stats(), *67*, *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143*, *145–148*, *150–153*
espn_wbb_teams, 144
espn_wbb_teams(), *67*, *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143, 144*, *146–148*, *150–153*
espn_wbb_tournament, 145
espn_wbb_tournament(), *70–73*, *75–79*, *81, 82*, *84*, *86–90*, *92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–145*, *147–153*
espn_wbb_tournament_season, 146
espn_wbb_tournament_season(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–146*, *148*, *150–153*
espn_wbb_tournament_seasons, 147
espn_wbb_tournament_seasons(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–147*,

*150–153*
espn_wbb_tournaments, 149
espn_wbb_tournaments(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–148*, *151–153*
espn_wbb_venues, 150
espn_wbb_venues(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–148*, *150*, *152, 153*
espn_wbb_week_ranking, 151
espn_wbb_week_ranking(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–148*, *150–153*
espn_wbb_week_rankings, 152
espn_wbb_week_rankings(), *70–73*, *75–79*, *81, 82*, *84*, *86–89*, *91, 92*, *94–97*, *99–102*, *104*, *107*, *109–116*, *119, 120*, *122–125*, *127–130*, *132, 133*, *135–137*, *139–141*, *143–148*, *150–152*
espn_wnba *(*espn_basketball*)*, 66
espn_wnba_athletes_index, 154
espn_wnba_athletes_index(), *156–158*, *160–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_award, 155
espn_wnba_award(), *155*, *157, 158*, *160–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_calendar, 156
espn_wnba_calendar(), *155, 156*, *158*, *160–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_coach_season, 157
espn_wnba_coach_season(), *155–157*, *160–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_coaches, 159
espn_wnba_coaches(), *155–158*, *161–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_conferences, 160
espn_wnba_conferences(), *155–158*, *160*, *162–164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_draft, 161
espn_wnba_draft(), *155–158*, *160, 161*, *163, 164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_draft_athlete_detail, 162
espn_wnba_draft_athlete_detail(), *155–158*, *160–162*, *164*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_draft_athletes

(espn_wnba_draft_rounds), 165
espn_wnba_draft_athletes(), *162*, *165*
espn_wnba_draft_pick, 164
espn_wnba_draft_pick(), *155–158*, *160–163*, *166–168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_draft_rounds, 165
espn_wnba_draft_rounds(), *155–158*, *160–165*, *167, 168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_draft_status (espn_wnba_draft_rounds), 165
espn_wnba_draft_status(), *165*
espn_wnba_franchise, 167
espn_wnba_franchise(), *155–158*, *160–164*, *166*, *168*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_franchises, 168
espn_wnba_franchises(), *155–158*, *160–164*, *166, 167*, *170–172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_freeagents, 169
espn_wnba_freeagents(), *155–158*, *160–164*, *166–168*, *171, 172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_futures, 170
espn_wnba_futures(), *155–158*, *160–164*, *166–168*, *170*, *172*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210, 211*, *213–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_game_all, 171
espn_wnba_game_all(), *67*, *155–158*, *160–164*, *166–168*, *170, 171*, *174*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210*, *212–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_game_broadcasts (espn_wnba_game_odds), 173
espn_wnba_game_odds, 173
espn_wnba_game_odds(), *155–158*, *160–164*, *166–168*, *170–172*, *176–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210*, *212–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_game_official_detail, 175
espn_wnba_game_official_detail(), *155–158*, *160–164*, *166–168*, *170–172*, *174*, *177, 178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210*, *212–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_game_officials (espn_wnba_game_odds), 173
espn_wnba_game_officials(), *175*
espn_wnba_game_play, 176
espn_wnba_game_play(), *155–158*, *160–164*, *166–168*, *170–172*, *174*, *176*, *178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210*, *212–215*, *217–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_game_play_personnel, 178
espn_wnba_game_play_personnel(), *155–158*, *160–164*, *166–168*, *170–172*, *174*, *176, 177*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*,

198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_player_box, 179
espn_wnba_game_player_box(), 155–158, 160–164, 166–168, 170–172, 175–178, 181, 182, 184, 185, 187–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_powerindex (espn_wnba_game_odds), 173
espn_wnba_game_predictor (espn_wnba_game_odds), 173
espn_wnba_game_probabilities, 180
espn_wnba_game_probabilities(), 155–158, 160–164, 166–168, 170–172, 175–178, 180, 182, 184, 185, 187–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_propbets, 181
espn_wnba_game_propbets(), 155–158, 160–164, 166–168, 170–172, 175–178, 180, 181, 184, 185, 187–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_rosters (espn_wnba_game_all), 171
espn_wnba_game_rosters(), 67
espn_wnba_game_situation (espn_wnba_game_odds), 173
espn_wnba_game_team_leaders (espn_wnba_game_team_linescores), 183
espn_wnba_game_team_linescores, 183
espn_wnba_game_team_linescores(), 155–158, 160–164, 166–168, 170–172, 175–178, 180–183, 185, 187–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_team_records (espn_wnba_game_team_linescores), 183
espn_wnba_game_team_roster (espn_wnba_game_team_linescores), 183
espn_wnba_game_team_roster(), 185
espn_wnba_game_team_roster_entry, 185
espn_wnba_game_team_roster_entry(), 155–158, 160–164, 166–168, 170–172, 175–178, 180–182, 184, 187–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–215, 217–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_game_team_score (espn_wnba_game_team_linescores), 183
espn_wnba_game_team_statistics (espn_wnba_game_team_linescores), 183
espn_wnba_game_team_statistics(), 179
espn_wnba_injuries, 186
espn_wnba_injuries(), 155–158, 160–164, 166–168, 170–172, 175–178, 180–182, 184, 185, 188–190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–214, 216–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_leaders, 187
espn_wnba_leaders(), 155–158, 160–163, 165–168, 170, 171, 173, 175–178, 180–182, 184, 185, 187, 189, 190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–214, 216–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_news, 188
espn_wnba_news(), 155–158, 160–163, 165–168, 170, 171, 173, 175–178, 180–182, 184, 185, 187, 188, 190, 192, 193, 195, 198–201, 203–206, 208, 210, 212–214, 216–219, 221–223, 225, 227–230, 232–234, 236–239, 241, 242
espn_wnba_pbp (espn_wnba_game_all), 171

espn_wnba_pbp(), *67*, *176*
espn_wnba_player_awards, 190
espn_wnba_player_awards(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184, 185*, *187–189*, *192, 193*, *195*, *198–201*, *203–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_player_box (espn_wnba_game_all), 171
espn_wnba_player_box(), *67*
espn_wnba_player_career_stats, 191
espn_wnba_player_career_stats(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184, 185*, *187–190*, *193*, *195*, *198–200*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_player_eventlog (espn_wnba_player_overview), 195
espn_wnba_player_eventlog(), *192*
espn_wnba_player_eventlog_v2, 192
espn_wnba_player_eventlog_v2(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184, 185*, *187–190*, *192*, *195*, *198–200*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_player_gamelog (espn_wnba_player_overview), 195
espn_wnba_player_info, 193
espn_wnba_player_info(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *198–200*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232–234*, *236–239*, *241, 242*
espn_wnba_player_overview, 195
espn_wnba_player_overview(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184, 185*, *187–190*, *192, 193*, *195*, *199*, *200*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232, 233*, *235–239*, *241, 242*
espn_wnba_player_seasons, 198
espn_wnba_player_seasons(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198*, *200*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232, 233*, *235–239*, *241, 242*
espn_wnba_player_splits (espn_wnba_player_overview), 195
espn_wnba_player_statisticslog (espn_wnba_player_overview), 195
espn_wnba_player_stats, 200
espn_wnba_player_stats(), *67*, *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *202–206*, *208*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–230*, *232, 233*, *235–239*, *241, 242*
espn_wnba_player_stats_v3 (espn_wnba_player_overview), 195
espn_wnba_position, 201
espn_wnba_position(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198–200*, *202–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_positions, 202
espn_wnba_positions(), *155–158*, *160–163*, *165–168*, *170, 171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198–200*, *202*, *204–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_powerindex, 203
espn_wnba_powerindex(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*,

*192, 193*, *195*, *198–200*, *202, 203*, *205, 206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_scoreboard, 204
espn_wnba_scoreboard(), *67*, *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198–200*, *202–204*, *206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_awards, 205
espn_wnba_season_awards(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198–200*, *202–205*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_draft *(*espn_wnba_draft_rounds*)*, 165
espn_wnba_season_group, 207
espn_wnba_season_group(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198–200*, *202–206*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_group_children *(*espn_wnba_season_group*)*, 207
espn_wnba_season_group_teams *(*espn_wnba_season_group*)*, 207
espn_wnba_season_groups, 209
espn_wnba_season_groups(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_info, 210
espn_wnba_season_info(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *213, 214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_leaders, 212
espn_wnba_season_leaders(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212*, *214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_ranking, 213
espn_wnba_season_ranking(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212, 213*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_rankings, 215
espn_wnba_season_rankings(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *217–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_type, 216
espn_wnba_season_type(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_types, 217
espn_wnba_season_types(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216, 217*, *219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241, 242*
espn_wnba_season_week, 218
espn_wnba_season_week(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–218*,

*221–223*, *225*, *227–229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_season_weeks, 220
espn_wnba_season_weeks(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *222*, *223*, *225*, *227–229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_seasons, 221
espn_wnba_seasons(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221*, *223*, *225*, *227–229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_standings, 222
espn_wnba_standings(), *67*, *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221*, *222*, *225*, *227–229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_team, 224
espn_wnba_team(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *227–229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_team_box (espn_wnba_game_all), 171
espn_wnba_team_box(), *67*
espn_wnba_team_injuries, 226
espn_wnba_team_injuries(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *228*, *229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_team_leaders (espn_wnba_team), 224
espn_wnba_team_news, 227
espn_wnba_team_news(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227*, *229*, *231–233*, *235–239*, *241*, *242*
espn_wnba_team_record, 228
espn_wnba_team_record(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227*, *228*, *230–239*, *241*, *242*
espn_wnba_team_record_detail, 230
espn_wnba_team_record_detail(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *232*, *233*, *235–239*, *241*, *242*
espn_wnba_team_roster (espn_wnba_team), 224
espn_wnba_team_roster(), *232*
espn_wnba_team_schedule, 231
espn_wnba_team_schedule(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231*, *233*, *235–239*, *241*, *242*
espn_wnba_team_season_profile (espn_wnba_team), 224
espn_wnba_team_season_roster, 232
espn_wnba_team_season_roster(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*, *184*, *186–190*, *192*, *193*, *195*, *198*, *199*, *201–206*, *209*, *210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231*, *232*, *235–239*, *241*, *242*
espn_wnba_team_season_statistics, 234
espn_wnba_team_season_statistics(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–178*, *180–182*,

*184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *236–239*, *241, 242*
espn_wnba_team_stats, 235
espn_wnba_team_stats(), *67*, *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235*, *237–239*, *241, 242*
espn_wnba_teams, 236
espn_wnba_teams(), *67*, *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235, 236*, *238, 239*, *241, 242*
espn_wnba_transactions, 237
espn_wnba_transactions(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–237*, *239*, *241, 242*
espn_wnba_venues, 238
espn_wnba_venues(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–238*, *241, 242*
espn_wnba_week_ranking, 240
espn_wnba_week_ranking(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *242*
espn_wnba_week_rankings, 241
espn_wnba_week_rankings(), *155–158*, *160–163*, *165–167*, *169–171*, *173*, *175–177*, *179–182*, *184*, *186–190*, *192, 193*, *195*, *198, 199*, *201–206*, *209, 210*, *212–214*, *216–219*, *221–223*, *225*, *227–229*, *231–233*, *235–239*, *241*

feb_game_info, 243
feb_game_info(), *246*, *248*, *250, 251*, *253*, *255*, *257*, *259, 260*, *263*
feb_leaders, 245
feb_leaders(), *244*, *248*, *250, 251*, *253*, *255*, *257*, *259, 260*, *263*
feb_overview, 246
feb_overview(), *243–246*, *248*, *250, 251*, *253*, *255*, *257*, *259–261*, *263*
feb_pbp, 248
feb_pbp(), *244*, *246*, *248*, *251*, *253*, *255*, *257*, *259, 260*, *263*
feb_period_scores, 250
feb_period_scores(), *244*, *246*, *248*, *250*, *253*, *255*, *257*, *259, 260*, *263*
feb_player_box, 251
feb_player_box(), *244*, *246*, *248*, *250, 251*, *255*, *257*, *259, 260*, *263*
feb_results, 253
feb_results(), *243–246*, *248*, *250, 251*, *253*, *257*, *259–261*, *263*
feb_schedule, 255
feb_schedule(), *243–246*, *248*, *250, 251*, *253*, *255*, *257*, *259–261*, *263*
feb_shots, 257
feb_shots(), *244*, *246*, *248*, *250, 251*, *253*, *255*, *257*, *260*, *263*
feb_standings, 259
feb_standings(), *244*, *246*, *248*, *250, 251*, *253*, *255*, *257*, *259*, *263*
feb_team_stats, 261
feb_team_stats(), *244*, *246*, *248*, *250, 251*, *253*, *255*, *257*, *259, 260*
fiba_ecw_competition, 263, *280*, *369*, *378*, *387*, *395*, *404*, *412*
fiba_ecw_editions *(*fiba_ecw_competition*)*, 263
fiba_ecw_pbp *(*fiba_ecw_competition*)*, 263
fiba_ecw_player_box *(*fiba_ecw_competition*)*, 263
fiba_ecw_schedule *(*fiba_ecw_competition*)*, 263
fiba_ecw_shots *(*fiba_ecw_competition*)*, 263

fiba_ecw_standings
(fiba_ecw_competition), 263
fiba_ecw_team_box
(fiba_ecw_competition), 263
fiba_ecw_teams (fiba_ecw_competition), 263
fiba_elw_competition, 271, *271*, *369*, *378*, *387*, *395*, *404*, *412*
fiba_elw_editions
(fiba_elw_competition), 271
fiba_elw_pbp (fiba_elw_competition), 271
fiba_elw_player_box
(fiba_elw_competition), 271
fiba_elw_schedule
(fiba_elw_competition), 271
fiba_elw_shots (fiba_elw_competition), 271
fiba_elw_standings
(fiba_elw_competition), 271
fiba_elw_team_box
(fiba_elw_competition), 271
fiba_elw_teams (fiba_elw_competition), 271
fiba_gamepage_box_players, 280
fiba_gamepage_box_players(), *268*, *276*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*, *366*, *374*, *383*, *392*, *400*, *409*
fiba_gamepage_box_teams, 283
fiba_gamepage_box_teams(), *269*, *277*, *282*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*, *367*, *375*, *384*, *393*, *401*, *410*
fiba_gamepage_lineup, 286
fiba_gamepage_lineup(), *282*, *285*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gamepage_pbp, 288
fiba_gamepage_pbp(), *270*, *279*, *282*, *285*, *287*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*, *368*, *377*, *385*, *394*, *402*, *411*
fiba_gamepage_period_scores, 290
fiba_gamepage_period_scores(), *282*, *285*, *287*, *290*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gamepage_shots, 292
fiba_gamepage_shots(), *271*, *279*, *282*, *285*, *287*, *290*, *292*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*, *369*, *377*, *386*, *395*, *403*, *412*
fiba_gdap_awards, 294
fiba_gdap_awards(), *264*, *272*, *282*, *285*, *287*, *290*, *292*, *294*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *326*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361, 362*, *371*, *379*, *388*, *396*, *405*
fiba_gdap_competition, 296
fiba_gdap_competition(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_editions, 298
fiba_gdap_editions(), *264*, *273*, *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *303*, *305*, *307*, *309*, *311*, *313*, *316, 317*, *319*, *321, 322*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343, 344*, *348*, *351*, *354*, *356*, *359*, *361, 362*, *371*, *379*, *388*, *396*, *405*
fiba_gdap_federation_games, 301
fiba_gdap_federation_games(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *305*, *307*, *309*, *311*, *313*, *316, 317*,

*319*, *321*, *322*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_federation_ranking, 304
fiba_gdap_federation_ranking(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *322*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_final_standings, 306
fiba_gdap_final_standings(), *267*, *275*, *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*, *365*, *373*, *382*, *391*, *399*, *408*
fiba_gdap_game, 307
fiba_gdap_game(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_game_officials, 310
fiba_gdap_game_officials(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_game_periods, 311
fiba_gdap_game_periods(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_games_between, 313
fiba_gdap_games_between(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_gateway_competition_ids, 316
fiba_gdap_gateway_competition_ids(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_gateway_federation_ranking, 318
fiba_gdap_gateway_federation_ranking(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *344*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_medal_count, 320
fiba_gdap_medal_count(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *303*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_national_teams, 321
fiba_gdap_national_teams(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_nominations, 323
fiba_gdap_nominations(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_player_game_stats, 325
fiba_gdap_player_game_stats(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_player_info, 327
fiba_gdap_player_info(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *331*, *333*,

*335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_player_stats, 329
fiba_gdap_player_stats(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_ranking_zones, 332
fiba_gdap_ranking_zones(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_roster, 333
fiba_gdap_roster(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_roster_coaches, 335
fiba_gdap_roster_coaches(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_roster_latest, 337
fiba_gdap_roster_latest(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_round_gamedays, 339
fiba_gdap_round_gamedays(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *309*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *343*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_round_standings, 341
fiba_gdap_round_standings(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *331*, *333*, *335*, *337*, *339*, *341*, *345*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_rounds, 343
fiba_gdap_rounds(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *348*, *351*, *354*, *356*, *359*, *361*
fiba_gdap_schedule, 345
fiba_gdap_schedule(), *265*, *273*, *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *351*, *354*, *356*, *359*, *361*, *363*, *372*, *380*, *389*, *397*, *406*
fiba_gdap_team_games, 348
fiba_gdap_team_games(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *354*, *356*, *359*, *361*
fiba_gdap_team_player_stats, 351
fiba_gdap_team_player_stats(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *356*, *359*, *361*
fiba_gdap_team_staff, 354
fiba_gdap_team_staff(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *359*, *361*
fiba_gdap_team_stats, 356
fiba_gdap_team_stats(), *282*, *285*, *287*, *290*, *292*, *294*, *296*, *298*, *300*, *304*, *305*, *307*, *310*, *311*, *313*, *316*, *317*, *319*, *321*, *323*, *324*, *327*, *329*, *332*, *333*, *335*, *337*, *339*, *341*, *343*, *345*, *348*, *351*, *354*, *356*, *361*
fiba_gdap_teams, 359
fiba_gdap_teams(), *267*, *275*, *282*, *285*, *287*,

290, 292, 294, 296, 298, 300, 304, 305, 307, 310, 311, 313, 316, 317, 319, 321, 323, 324, 327, 329, 332, 333, 335, 337, 339, 341, 343, 345, 348, 351, 354, 356, 359, 365, 374, 382, 391, 399, 408
fiba_wafrobasket_competition, 271, 280, 361, 378, 387, 395, 404, 412
fiba_wafrobasket_editions (fiba_wafrobasket_competition), 361
fiba_wafrobasket_pbp (fiba_wafrobasket_competition), 361
fiba_wafrobasket_player_box (fiba_wafrobasket_competition), 361
fiba_wafrobasket_schedule (fiba_wafrobasket_competition), 361
fiba_wafrobasket_shots (fiba_wafrobasket_competition), 361
fiba_wafrobasket_standings (fiba_wafrobasket_competition), 361
fiba_wafrobasket_team_box (fiba_wafrobasket_competition), 361
fiba_wafrobasket_teams (fiba_wafrobasket_competition), 361
fiba_wamericup_competition, 271, 280, 369, 370, 387, 395, 404, 412
fiba_wamericup_editions (fiba_wamericup_competition), 370
fiba_wamericup_pbp (fiba_wamericup_competition), 370
fiba_wamericup_player_box (fiba_wamericup_competition), 370
fiba_wamericup_schedule (fiba_wamericup_competition), 370
fiba_wamericup_shots (fiba_wamericup_competition), 370
fiba_wamericup_standings (fiba_wamericup_competition), 370
fiba_wamericup_team_box (fiba_wamericup_competition), 370
fiba_wamericup_teams (fiba_wamericup_competition), 370
fiba_wasiacup_competition, 271, 280, 369, 378, 378, 395, 404, 412
fiba_wasiacup_editions (fiba_wasiacup_competition), 378
fiba_wasiacup_pbp (fiba_wasiacup_competition), 378
fiba_wasiacup_player_box (fiba_wasiacup_competition), 378
fiba_wasiacup_schedule (fiba_wasiacup_competition), 378
fiba_wasiacup_shots (fiba_wasiacup_competition), 378
fiba_wasiacup_standings (fiba_wasiacup_competition), 378
fiba_wasiacup_team_box (fiba_wasiacup_competition), 378
fiba_wasiacup_teams (fiba_wasiacup_competition), 378
fiba_weurobasket_competition, 271, 280, 369, 378, 387, 387, 404, 412
fiba_weurobasket_editions (fiba_weurobasket_competition), 387
fiba_weurobasket_pbp (fiba_weurobasket_competition), 387
fiba_weurobasket_player_box (fiba_weurobasket_competition), 387
fiba_weurobasket_schedule

(fiba_weurobasket_competition), 387
fiba_weurobasket_shots (fiba_weurobasket_competition), 387
fiba_weurobasket_standings (fiba_weurobasket_competition), 387
fiba_weurobasket_team_box (fiba_weurobasket_competition), 387
fiba_weurobasket_teams (fiba_weurobasket_competition), 387
fiba_woly_competition, *271*, *280*, *369*, *378*, *387*, 395, *395*, *412*
fiba_woly_editions (fiba_woly_competition), 395
fiba_woly_pbp (fiba_woly_competition), 395
fiba_woly_player_box (fiba_woly_competition), 395
fiba_woly_schedule (fiba_woly_competition), 395
fiba_woly_shots (fiba_woly_competition), 395
fiba_woly_standings (fiba_woly_competition), 395
fiba_woly_team_box (fiba_woly_competition), 395
fiba_woly_teams (fiba_woly_competition), 395
fiba_wwc_competition, *271*, *280*, *369*, *378*, *387*, *395*, 404, *404*
fiba_wwc_editions (fiba_wwc_competition), 404
fiba_wwc_pbp (fiba_wwc_competition), 404
fiba_wwc_player_box (fiba_wwc_competition), 404
fiba_wwc_schedule (fiba_wwc_competition), 404
fiba_wwc_shots (fiba_wwc_competition), 404
fiba_wwc_standings (fiba_wwc_competition), 404
fiba_wwc_team_box (fiba_wwc_competition), 404
fiba_wwc_teams (fiba_wwc_competition), 404
fox_basketball_boxscore (fox_wnba_pbp), 414
fox_basketball_league_leaders, 412, *414–416*
fox_basketball_standings (fox_wnba_team_roster), 415
fox_wbb_boxscore (fox_wnba_pbp), 414
fox_wbb_league_leaders (fox_basketball_league_leaders), 412
fox_wbb_odds (fox_wnba_pbp), 414
fox_wbb_pbp (fox_wnba_pbp), 414
fox_wbb_standings (fox_wnba_team_roster), 415
fox_wbb_team_gamelog (fox_wnba_team_roster), 415
fox_wbb_team_roster (fox_wnba_team_roster), 415
fox_wbb_team_stats (fox_wnba_team_roster), 415
fox_wbb_teams (fox_wnba_team_roster), 415
fox_wbb_teams_all, 413
fox_wbb_teams_all(), *413*, *415*, *416*
fox_wnba_boxscore (fox_wnba_pbp), 414
fox_wnba_league_leaders (fox_basketball_league_leaders), 412
fox_wnba_odds (fox_wnba_pbp), 414
fox_wnba_pbp, 414
fox_wnba_pbp(), *413*, *414*, *416*
fox_wnba_standings (fox_wnba_team_roster), 415
fox_wnba_team_gamelog (fox_wnba_team_roster), 415
fox_wnba_team_roster, 415
fox_wnba_team_roster(), *413–415*
fox_wnba_team_stats (fox_wnba_team_roster), 415
fox_wnba_teams (fox_wnba_team_roster), 415

genius_competition_games, 417
genius_competition_games(), *420–429*, *431*, *433–435*, *437*, *438*, *440*, *450*, *690*
genius_competitions, 419

genius_competitions(), *417–440*, *445, 446*, *450*, *684, 685*, *690*
genius_game, 421
genius_game(), *419, 420*, *423*, *425*, *427*, *429*, *431*, *433*, *435*, *437, 438*, *440*, *446*, *450*, *685*, *690*
genius_game_crosswalk, 422
genius_game_crosswalk(), *419, 420*, *422*, *425*, *427*, *429*, *431*, *433*, *435*, *437, 438*, *440*, *450*, *690*
genius_match_competition, 424
genius_match_competition(), *418–424*, *426–440*, *450*, *690*
genius_pbp, 425
genius_pbp(), *419, 420*, *422, 423*, *425*, *429*, *431*, *433*, *435*, *437, 438*, *440*, *445*, *449, 450*, *684*, *689, 690*
genius_player_box, 427
genius_player_box(), *419, 420*, *422, 423*, *425*, *427*, *431*, *433*, *435*, *437, 438*, *440*, *445*, *448*, *450*, *684*, *688*, *690*
genius_schedule, 430
genius_schedule(), *419–429*, *432–435*, *438*, *440*, *445, 446*, *450*, *684*, *686*, *690*
genius_schedule_embed, 432
genius_schedule_embed(), *419, 420*, *422, 423*, *425*, *427*, *429*, *431*, *435*, *438*, *440*, *450*, *690*
genius_shots, 433
genius_shots(), *419, 420*, *422, 423*, *425*, *427*, *429*, *431*, *433*, *438*, *440*, *445*, *450*, *684*, *690*
genius_team_box, 435
genius_team_box(), *419, 420*, *422, 423*, *425*, *427*, *429*, *431*, *433*, *435*, *438*, *440*, *445*, *447*, *450*, *684*, *686*, *690*
genius_team_crosswalk, 438
genius_team_crosswalk(), *419, 420*, *422, 423*, *425*, *427*, *429*, *431*, *433*, *435*, *438*, *440*, *450*, *690*
genius_widget_games, 439
genius_widget_games(), *419–429*, *431*, *433–435*, *438*, *450*, *690*

helper_espn_wbb_pbp, 440
helper_espn_wbb_player_box *(*helper_espn_wbb_pbp*)*, 440
helper_espn_wbb_team_box *(*helper_espn_wbb_pbp*)*, 440
helper_espn_wnba_pbp, 441
helper_espn_wnba_player_box *(*helper_espn_wnba_pbp*)*, 441
helper_espn_wnba_team_box *(*helper_espn_wnba_pbp*)*, 441
hhs_team_roster, 442
hhs_team_roster(), *443, 444*
hhs_team_stats, 443
hhs_team_stats(), *442*, *444*
hhs_teams, 444
hhs_teams(), *442, 443*

janitor::make_clean_names(), *106*, *197*

lfb_competition, *419, 420*, *422, 423*, *425*, *427*, *429*, *431*, *433*, *435*, *438*, *440*, 445, *690*
lfb_competitions *(*lfb_competition*)*, 445
lfb_pbp *(*lfb_competition*)*, 445
lfb_player_box *(*lfb_competition*)*, 445
lfb_schedule *(*lfb_competition*)*, 445
lfb_shots *(*lfb_competition*)*, 445
lfb_team_box *(*lfb_competition*)*, 445
lmb_competition, *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *37*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, 451, *464*, *504*, *567*
lmb_pbp *(*lmb_competition*)*, 451
lmb_player_box *(*lmb_competition*)*, 451
lmb_schedule *(*lmb_competition*)*, 451
lmb_seasons *(*lmb_competition*)*, 451
lmb_shots *(*lmb_competition*)*, 451
lmb_standings *(*lmb_competition*)*, 451
lmb_team_box *(*lmb_competition*)*, 451
lmb_teams *(*lmb_competition*)*, 451
lnbp_competition, *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *37*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, 457, *457*, *504*, *567*
lnbp_pbp *(*lnbp_competition*)*, 457
lnbp_player_box *(*lnbp_competition*)*, 457
lnbp_schedule *(*lnbp_competition*)*, 457
lnbp_seasons *(*lnbp_competition*)*, 457
lnbp_shots *(*lnbp_competition*)*, 457
lnbp_standings *(*lnbp_competition*)*, 457
lnbp_team_box *(*lnbp_competition*)*, 457
lnbp_teams *(*lnbp_competition*)*, 457
load_ncaa_wbb_lineups *(*load_ncaa_wbb_pbp*)*, 464
load_ncaa_wbb_matchup_stints *(*load_ncaa_wbb_pbp*)*, 464

load_ncaa_wbb_pbp, 464
load_ncaa_wbb_player_box (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_possessions (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_rapm (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_rapm_within_team (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_rapm_within_team(), *468*
load_ncaa_wbb_rosters (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_schedule (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_shots (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_team_box (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_team_ids (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_team_rosters (load_ncaa_wbb_pbp), 464
load_ncaa_wbb_team_rosters(), *465*
load_wbb (wehoop_loaders), 690
load_wbb_game_rosters (load_wbb_pbp), 473
load_wbb_game_rosters_manifest (load_wbb_pbp), 473
load_wbb_group_aliases, 470
load_wbb_group_aliases(), *472*, *473*, *479*
load_wbb_group_seasons, 471
load_wbb_group_seasons(), *471–473*, *479*, *683*
load_wbb_groups, 472
load_wbb_groups(), *471*, *472*, *479*
load_wbb_officials (load_wbb_pbp), 473
load_wbb_officials_manifest (load_wbb_pbp), 473
load_wbb_pbp, 473
load_wbb_pbp(), *691*
load_wbb_player_box (load_wbb_pbp), 473
load_wbb_player_box(), *691*
load_wbb_player_core (load_wbb_pbp), 473
load_wbb_player_crosswalk (load_wnba_team_crosswalk), 495
load_wbb_player_stats (load_wbb_pbp), 473
load_wbb_player_stats_manifest (load_wbb_pbp), 473
load_wbb_player_value (load_wnba_player_impact), 485
load_wbb_ratings (load_wnba_player_impact), 485
load_wbb_rosters (load_wbb_pbp), 473
load_wbb_rosters_manifest (load_wbb_pbp), 473
load_wbb_schedule (load_wbb_pbp), 473
load_wbb_schedule(), *691*
load_wbb_schedule_crosswalk (load_wnba_team_crosswalk), 495
load_wbb_shots (load_wbb_pbp), 473
load_wbb_shots_manifest (load_wbb_pbp), 473
load_wbb_standings (load_wbb_pbp), 473
load_wbb_standings_manifest (load_wbb_pbp), 473
load_wbb_team_box (load_wbb_pbp), 473
load_wbb_team_box(), *691*
load_wbb_team_crosswalk (load_wnba_team_crosswalk), 495
load_wbb_team_group_seasons, 478
load_wbb_team_group_seasons(), *471–473*, *683*
load_wbb_team_stats (load_wbb_pbp), 473
load_wbb_team_stats_manifest (load_wbb_pbp), 473
load_wnba (wehoop_loaders), 690
load_wnba_draft (load_wnba_pbp), 480
load_wnba_draft_manifest (load_wnba_pbp), 480
load_wnba_game_rosters (load_wnba_pbp), 480
load_wnba_game_rosters_manifest (load_wnba_pbp), 480
load_wnba_group_aliases (load_wbb_group_aliases), 470
load_wnba_group_seasons (load_wbb_group_seasons), 471
load_wnba_groups (load_wbb_groups), 472
load_wnba_officials (load_wnba_pbp), 480
load_wnba_officials_manifest (load_wnba_pbp), 480
load_wnba_pbp, 480
load_wnba_pbp(), *690*
load_wnba_player_box (load_wnba_pbp), 480

load_wnba_player_box(), *690*
load_wnba_player_core *(*load_wnba_pbp*)*, 480
load_wnba_player_crosswalk *(*load_wnba_team_crosswalk*)*, 495
load_wnba_player_impact, 485
load_wnba_player_stats *(*load_wnba_pbp*)*, 480
load_wnba_player_stats_manifest *(*load_wnba_pbp*)*, 480
load_wnba_rosters *(*load_wnba_pbp*)*, 480
load_wnba_rosters_manifest *(*load_wnba_pbp*)*, 480
load_wnba_schedule *(*load_wnba_pbp*)*, 480
load_wnba_schedule(), *690*
load_wnba_schedule_crosswalk *(*load_wnba_team_crosswalk*)*, 495
load_wnba_shots *(*load_wnba_pbp*)*, 480
load_wnba_shots_manifest *(*load_wnba_pbp*)*, 480
load_wnba_standings *(*load_wnba_pbp*)*, 480
load_wnba_standings_manifest *(*load_wnba_pbp*)*, 480
load_wnba_stats_coaches *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_coaches_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_draft *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_draft_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_game_rosters *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_game_rosters_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_leaguedash, 487
load_wnba_stats_leaguedash(), *488*, *489*, *494*
load_wnba_stats_lineups *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_lineups(), *487*
load_wnba_stats_lineups_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_officials *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_officials_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_pbp *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_pbp_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_player_game_logs *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_player_game_logs(), *489*
load_wnba_stats_player_game_logs_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_player_stats *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_player_stats(), *487*
load_wnba_stats_player_stats_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_possessions *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_rosters, 488
load_wnba_stats_rosters(), *488*
load_wnba_stats_rosters_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_schedule *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_schedule_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_shots *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_shots_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_standings *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_standings(), *487*
load_wnba_stats_standings_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_team_stats *(*load_wnba_stats_rosters*)*, 488
load_wnba_stats_team_stats(), *487*
load_wnba_stats_team_stats_manifest *(*load_wnba_stats_rosters*)*, 488
load_wnba_team_box *(*load_wnba_pbp*)*, 480
load_wnba_team_box(), *690*
load_wnba_team_crosswalk, 495
load_wnba_team_crosswalk(), *682–684*, *813*, *860*, *875*
load_wnba_team_group_seasons *(*load_wbb_team_group_seasons*)*, 478
load_wnba_team_stats *(*load_wnba_pbp*)*, 480

load_wnba_team_stats_manifest (load_wnba_pbp), 480
lpb_competition, *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *37*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, 497, *567*
lpb_pbp (lpb_competition), 497
lpb_player_box (lpb_competition), 497
lpb_schedule (lpb_competition), 497
lpb_seasons (lpb_competition), 497
lpb_shots (lpb_competition), 497
lpb_standings (lpb_competition), 497
lpb_team_box (lpb_competition), 497
lpb_teams (lpb_competition), 497

most_recent_wbb_season, 504
most_recent_wnba_season, 504
most_recent_wnba_season(), *505*
most_recent_wnba_stats_season, 505

nbl1_competition, *14*, *16–18*, *21*, *23*, *25*, *28*, *30*, *32*, *34*, *37*, *39*, *41*, *43–45*, *47*, *50*, *52*, *55*, *57*, *59*, *61*, *457*, *464*, *504*, 560
nbl1_pbp (nbl1_competition), 560
nbl1_player_box (nbl1_competition), 560
nbl1_schedule (nbl1_competition), 560
nbl1_seasons (nbl1_competition), 560
nbl1_shots (nbl1_competition), 560
nbl1_standings (nbl1_competition), 560
nbl1_team_box (nbl1_competition), 560
nbl1_teams (nbl1_competition), 560
nbl_rosetta_match, 505
nbl_rosetta_match(), *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_match_by_id, 508
nbl_rosetta_match_by_id(), *507*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_match_pbp, 510
nbl_rosetta_match_pbp(), *507*, *510*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_match_player_box, 512
nbl_rosetta_match_player_box(), *507*, *510*, *512*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_match_player_period_box, 515
nbl_rosetta_match_player_period_box(), *507*, *510*, *512*, *515*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_match_team_box, 517
nbl_rosetta_match_team_box(), *507*, *510*, *512*, *515*, *517*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_matches, 519
nbl_rosetta_matches(), *507*, *510*, *512*, *515*, *517*, *519*, *525*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_next_match, 523
nbl_rosetta_next_match(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *528*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_next_match_team, 525
nbl_rosetta_next_match_team(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *530*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_next_matches, 528
nbl_rosetta_next_matches(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *532*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_outcomes, 530
nbl_rosetta_outcomes(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *535*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_player_game_log, 532
nbl_rosetta_player_game_log(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *538*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_player_stats, 535
nbl_rosetta_player_stats(), *507*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *541, 542*, *544*, *546*, *550*, *552*, *555*, *558*, *560*

nbl_rosetta_players, 539
nbl_rosetta_players(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *543*, *544*, *547*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_seasons, 541
nbl_rosetta_seasons(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *544*, *547*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_seasons_current, 543
nbl_rosetta_seasons_current(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *547*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_standings, 545
nbl_rosetta_standings(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *550*, *552*, *555*, *558*, *560*
nbl_rosetta_stats_leaders, 547
nbl_rosetta_stats_leaders(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *547*, *552*, *555*, *558*, *560*
nbl_rosetta_team_players, 550
nbl_rosetta_team_players(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *547*, *550*, *555*, *558*, *560*
nbl_rosetta_team_stats, 552
nbl_rosetta_team_stats(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *547*, *550*, *552*, *558*, *560*
nbl_rosetta_team_stats_leaders, 555
nbl_rosetta_team_stats_leaders(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *547*, *550*, *552*, *555*, *560*
nbl_rosetta_teams, 558
nbl_rosetta_teams(), *508*, *510*, *512*, *515*, *517*, *519*, *522*, *525*, *528*, *530*, *532*, *535*, *538*, *541*, *543*, *544*, *547*, *550*, *552*, *555*, *558*
ncaa *(*ncaa_wbb*)*, 567
ncaa_wbb, 567
ncaa_wbb_NET_rankings, 568
ncaa_wbb_NET_rankings(), *567*
ncaa_wbb_teams, 568
ncaa_wbb_teams(), *567*
ncaa_womens *(*ncaa_wbb*)*, 567

parameter_descriptions, 569
parquet_from_url, 570
progressively, 570

rds_from_url, 571
rds_from_url(), *570*
readRDS(), *571*

scb_game, 571
scb_game(), *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_game_actions, 575
scb_game_actions(), *574*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_game_players, 577
scb_game_players(), *574*, *576*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_game_teams, 579
scb_game_teams(), *574*, *576*, *579*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_games, 582
scb_games(), *574*, *576*, *579*, *582*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_games_recent, 586
scb_games_recent(), *574*, *576*, *579*, *582*, *585*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_player_statistics, 589
scb_player_statistics(), *574*, *576*, *579*, *582*, *585*, *589*, *594*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_playoff_games, 592

scb_playoff_games(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *596*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_playoff_series, 594
scb_playoff_series(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *597*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *613*, *615*, *618*, *619*, *621*
scb_records_per_game, 596
scb_records_per_game(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_records_season_averages, 598
scb_records_season_averages(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_referees, 599
scb_referees(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_season_player, 601
scb_season_player(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_season_players, 603
scb_season_players(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *597*, *599*, *601*, *603*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_season_teams, 604
scb_season_teams(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *607*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_seasons, 606
scb_seasons(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *610*, *612*, *614*, *615*, *618*, *619*, *621*
scb_standings, 608
scb_standings(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *612*, *614*, *615*, *618*, *619*, *621*
scb_team, 610
scb_team(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *614*, *615*, *618*, *619*, *621*
scb_team_rosters, 612
scb_team_rosters(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *615*, *618*, *619*, *621*
scb_team_seasons, 614
scb_team_seasons(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *618*, *619*, *621*
scb_team_statistics, 615
scb_team_statistics(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *619*, *621*
scb_transfers, 618
scb_transfers(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *621*
scb_transfers_teams, 620
scb_transfers_teams(), *574*, *576*, *579*, *582*, *585*, *589*, *592*, *594*, *596*, *598*, *599*, *601*, *603*, *604*, *606*, *607*, *610*, *612*, *614*, *615*, *618*, *619*

tbf_daily_matches, 621
tbf_daily_matches(), *625*, *627*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *660*, *662*, *664*, *666*, *668*, *670*, *672*, *674*, *676*, *677*
tbf_game_header, 623
tbf_game_header(), *623*, *627*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *660*, *662*, *664*, *666*, *668*, *670*, *672*, *674*, *676*, *677*
tbf_game_leaders, 625
tbf_game_leaders(), *623*, *625*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *660*, *662*, *664*, *666*,

668, 670, 672, 674, 676, 677
tbf_game_players, 627
tbf_game_players(), 623, 625, 627, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 660, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_highlights, 628
tbf_highlights(), 623, 625, 627, 628, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 660, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_assists, 630
tbf_leaders_assists(), 623, 625, 627, 628, 630, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 660, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_blocks, 632
tbf_leaders_blocks(), 623, 625, 627, 628, 630, 631, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 660, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_efficiency, 633
tbf_leaders_efficiency(), 623, 625, 627, 628, 630, 631, 633, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 660, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_minutes, 635
tbf_leaders_minutes(), 623, 625, 627, 628, 630, 631, 633, 635, 638, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_points, 637
tbf_leaders_points(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 640, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_rebounds, 638
tbf_leaders_rebounds(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 641, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_steals, 640
tbf_leaders_steals(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 643, 645, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_threes, 642
tbf_leaders_threes(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 645, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leaders_turnovers, 643
tbf_leaders_turnovers(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 646, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_league_seasons, 645
tbf_league_seasons(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 648, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_leagues, 647
tbf_leagues(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 650, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_pbp, 648
tbf_pbp(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 651, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_period_scores, 650
tbf_period_scores(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 653, 655, 657, 659, 661, 662, 664, 666, 668, 670, 672, 674, 676, 677
tbf_player_box, 651
tbf_player_box(), 623, 625, 627, 628, 630, 631, 633, 635, 636, 638, 640, 641, 643, 645, 646, 648, 650, 651, 655, 657, 659, 661, 662, 664, 666, 668,

*670*, *672*, *674*, *676*, *677*
tbf_player_career, 654
tbf_player_career(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *657*, *659*, *661*, *662*, *664*, *666*, *668*, *670*, *672*, *674*, *676*, *677*
tbf_players, 656
tbf_players(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653–655*, *659*, *661*, *662*, *664*, *666*, *668*, *670*, *672*, *674*, *676*, *677*
tbf_schedule, 657
tbf_schedule(), *623–631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648–653*, *655*, *657*, *661*, *662*, *664–669*, *671–674*, *676*, *678*
tbf_seasons, 659
tbf_seasons(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *636*, *638*, *640*, *641*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *662*, *664*, *666*, *668*, *671*, *672*, *674*, *676*, *678*
tbf_shots, 661
tbf_shots(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *664*, *666*, *668*, *671*, *672*, *674*, *676*, *678*
tbf_standings, 662
tbf_standings(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *666*, *668*, *671*, *672*, *674*, *676*, *678*
tbf_team_box, 664
tbf_team_box(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *668*, *671*, *672*, *674*, *676*, *678*
tbf_team_games, 666
tbf_team_games(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *666*, *671*, *672*, *674*, *676*, *678*
tbf_team_player_stats, 668
tbf_team_player_stats(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *666*, *668*, *672*, *674*, *676*, *678*
tbf_team_roster, 671
tbf_team_roster(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *666*, *668*, *671*, *674*, *676*, *678*
tbf_team_staff, 673
tbf_team_staff(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *666*, *668*, *671*, *672*, *676*, *678*
tbf_teams, 674
tbf_teams(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657*, *659*, *661*, *662*, *664*, *666–669*, *671–674*, *678*
tbf_weeks, 676
tbf_weeks(), *623*, *625*, *627*, *628*, *630*, *631*, *633*, *635*, *637*, *638*, *640*, *642*, *643*, *645*, *646*, *648*, *650*, *651*, *653*, *655*, *657–659*, *661–664*, *666*, *668*, *671*, *672*, *674*, *676*

update_wbb_db, 678
update_wbb_db(), *691*
update_wnba_db, 679
update_wnba_db(), *690*
update_wnba_stats_db, 681

wbb_player_crosswalk, 682
wbb_player_crosswalk(), *496*, *683*, *684*
wbb_schedule_crosswalk, 682
wbb_schedule_crosswalk(), *496*, *682*, *684*
wbb_team_crosswalk, 683
wbb_team_crosswalk(), *496*, *682*, *683*
wbbl_competition, *419*, *420*, *422*, *423*, *425*, *427*, *429*, *431*, *433*, *435*, *438*, *440*, *450*, 684
wbbl_competitions *(*wbbl_competition*)*, 684
wbbl_pbp *(*wbbl_competition*)*, 684

wbbl_player_box *(wbbl_competition)*, 684
wbbl_schedule *(wbbl_competition)*, 684
wbbl_shots *(wbbl_competition)*, 684
wbbl_team_box *(wbbl_competition)*, 684
wehoop_loaders, 690
wehoop_wnba_stats *(wnba_stats)*, 873
wjbl_box_score, 691
wjbl_box_score(), *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_game, 693
wjbl_game(), *693*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_game_info, 695
wjbl_game_info(), *693*, *695*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_game_months, 698
wjbl_game_months(), *693*, *695*, *698*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_head_to_head, 700
wjbl_head_to_head(), *693*, *695*, *698*, *700*, *703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_leagues, 702
wjbl_leagues(), *693*, *695*, *698*, *700*, *702*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_pbp, 704
wjbl_pbp(), *693*, *695*, *698*, *700*, *702, 703*, *707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_period_scores, 706
wjbl_period_scores(), *693*, *695*, *698*, *700*, *702, 703*, *706*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_player, 708
wjbl_player(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_player_period_leaders, 710
wjbl_player_period_leaders(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_roster, 712
wjbl_roster(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *717*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_schedule, 714
wjbl_schedule(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *719*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_staff, 717
wjbl_staff(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *721*, *723*, *725*, *728*, *730, 731*
wjbl_standing_seasons, 719
wjbl_standing_seasons(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *723*, *725*, *728*, *730, 731*
wjbl_team_box_score, 721
wjbl_team_box_score(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *725*, *728*, *730, 731*
wjbl_team_period_stats, 723
wjbl_team_period_stats(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *728*, *730, 731*
wjbl_team_ranking, 725
wjbl_team_ranking(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *730, 731*
wjbl_team_standings, 728
wjbl_team_standings(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *731*
wjbl_teams, 730
wjbl_teams(), *693*, *695*, *698*, *700*, *702, 703*, *706, 707*, *710*, *712*, *714*, *717*, *719*, *721*, *723*, *725*, *728*, *730*
wnba_alltimeleadersgrids, 731
wnba_alltimeleadersgrids(), *737, 738*, *770*, *801*
wnba_assistleaders, 736
wnba_assistleaders(), *736*, *738*, *770*, *801*
wnba_assisttracker, 738
wnba_assisttracker(), *736, 737*, *770*, *801*
wnba_boxscoreadvancedv2 *(wnba_boxscoretraditionalv2)*, 740
wnba_boxscoreadvancedv2(), *874*

wnba_boxscoreadvancedv3 (wnba_boxscoretraditionalv3), 743
wnba_boxscoreadvancedv3(), *874*
wnba_boxscorefourfactorsv2 (wnba_boxscoretraditionalv2), 740
wnba_boxscorefourfactorsv2(), *874*
wnba_boxscorefourfactorsv3 (wnba_boxscoretraditionalv3), 743
wnba_boxscorefourfactorsv3(), *874*
wnba_boxscorehustlev2(), *739*, *747*
wnba_boxscoremiscv2 (wnba_boxscoretraditionalv2), 740
wnba_boxscoremiscv2(), *874*
wnba_boxscoremiscv3 (wnba_boxscoretraditionalv3), 743
wnba_boxscoremiscv3(), *874*
wnba_boxscoreplayertrackv2(), *874*
wnba_boxscoreplayertrackv3 (wnba_boxscoretraditionalv3), 743
wnba_boxscoreplayertrackv3(), *874*
wnba_boxscorescoringv2 (wnba_boxscoretraditionalv2), 740
wnba_boxscorescoringv2(), *874*
wnba_boxscorescoringv3 (wnba_boxscoretraditionalv3), 743
wnba_boxscorescoringv3(), *874*
wnba_boxscoresummaryv2(), *743*, *768*, *874*
wnba_boxscoresummaryv2_last_meeting_schema, *862*
wnba_boxscoresummaryv2_season_series_schema, *862*
wnba_boxscoresummaryv3, 739
wnba_boxscoresummaryv3(), *747*
wnba_boxscoretraditionalv2, 740
wnba_boxscoretraditionalv2(), *768*, *874*
wnba_boxscoretraditionalv3, 743
wnba_boxscoretraditionalv3(), *739*, *874*
wnba_boxscoreusagev2 (wnba_boxscoretraditionalv2), 740
wnba_boxscoreusagev2(), *874*
wnba_boxscoreusagev3 (wnba_boxscoretraditionalv3), 743
wnba_boxscoreusagev3(), *874*
wnba_commonallplayers, 748
wnba_commonallplayers(), *750*, *752*
wnba_commonplayerinfo, 749
wnba_commonplayerinfo(), *748*, *752*
wnba_commonplayoffseries, 750
wnba_commonteamroster, 751
wnba_commonteamroster(), *748*, *750*
wnba_cumestatsplayer, 752
wnba_cumestatsplayer(), *754, 755*, *757*
wnba_cumestatsplayergames, 753
wnba_cumestatsplayergames(), *753*, *755*, *757*
wnba_cumestatsteam, 754
wnba_cumestatsteam(), *753, 754*, *757*
wnba_cumestatsteamgames, 755
wnba_cumestatsteamgames(), *753–755*
wnba_data_pbp, 757
wnba_data_pbp(), *810–812*
wnba_draftboard, 758
wnba_draftboard(), *761, 762*, *874*
wnba_draftcombinedrillresults (wnba_draftcombinestats), 759
wnba_draftcombinenonstationaryshooting (wnba_draftcombinestats), 759
wnba_draftcombineplayeranthro (wnba_draftcombinestats), 759
wnba_draftcombinespotshooting (wnba_draftcombinestats), 759
wnba_draftcombinestats, 759
wnba_draftcombinestats(), *759*, *762*, *874*
wnba_drafthistory, 761
wnba_drafthistory(), *759*, *761*, *874*
wnba_fantasywidget, 762
wnba_fantasywidget(), *771*, *836*, *838*
wnba_franchisehistory, 764
wnba_franchisehistory(), *765–767*, *874*
wnba_franchiseleaders, 765
wnba_franchiseleaders(), *764*, *766, 767*, *874*
wnba_franchiseleaderswrank, 766
wnba_franchiseleaderswrank(), *764, 765*, *767*
wnba_franchiseplayers, 767

wnba_franchiseplayers(), *764–766*
wnba_gamerotation, 768
wnba_gamerotation(), *743*, *774*, *803*, *856, 857*, *865*, *873*, *889*
wnba_homepageleaders(), *874*
wnba_homepagewidget, 769
wnba_homepagewidget(), *736–738*, *801*
wnba_hustlestatsboxscore(), *874*
wnba_infographicfanduelplayer, 771
wnba_infographicfanduelplayer(), *763*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_leaderstiles(), *874*
wnba_leaguedashlineups, 772
wnba_leaguedashlineups(), *768*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *856, 857*, *865*, *869*, *873*, *889*
wnba_leaguedashplayerbiostats, 774
wnba_leaguedashplayerbiostats(), *771*, *774*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *869*
wnba_leaguedashplayerclutch, 777
wnba_leaguedashplayerclutch(), *771*, *774*, *776*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *869*
wnba_leaguedashplayershotlocations, 780
wnba_leaguedashplayershotlocations(), *771*, *774*, *776*, *779*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *814, 815*, *817*, *819*, *821*, *833–836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *868, 869*, *873*, *885*
wnba_leaguedashplayerstats, 783
wnba_leaguedashplayerstats(), *771*, *774*, *776*, *779*, *782*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *869*
wnba_leaguedashteamclutch, 786
wnba_leaguedashteamclutch(), *774*, *776*, *779*, *782*, *785*, *790*, *793*, *798, 799*, *803*, *805–807*, *834*, *869*, *885*
wnba_leaguedashteamclutch_league_dash_team_clutch_schema, *788*, *792*
wnba_leaguedashteamshotlocations, 788
wnba_leaguedashteamshotlocations(), *774*, *776*, *779*, *782*, *785*, *788*, *793*, *798, 799*, *803*, *805–807*, *834*, *868, 869*, *873*, *885*
wnba_leaguedashteamstats, 791
wnba_leaguedashteamstats(), *774*, *776*, *779*, *782*, *785*, *788*, *790*, *798, 799*, *803*, *805–807*, *869*, *885*, *889*, *891–893*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_leaguegamefinder, 793
wnba_leaguegamefinder(), *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *799*, *803*, *805–807*, *846*, *869*, *905*
wnba_leaguegamelog, 798
wnba_leaguegamelog(), *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798*, *803*, *805–807*, *869*
wnba_leagueleaders, 800
wnba_leagueleaders(), *736–738*, *770*, *874*
wnba_leaguelineupviz, 801
wnba_leaguelineupviz(), *768*, *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *805–807*, *856, 857*, *865*, *869*, *873*, *889*
wnba_leagueplayerondetails, 803
wnba_leagueplayerondetails(), *771*, *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *806, 807*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *869*
wnba_leagueseasonmatchups, 805
wnba_leagueseasonmatchups(), *771*, *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805*, *807*, *814, 815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *869*
wnba_leaguestandingsv3, 806
wnba_leaguestandingsv3(), *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*,

*803*, *805, 806*, *869*, *874*
wnba_live_boxscore, 808
wnba_live_pbp, 809
wnba_live_pbp(), *758*, *811, 812*, *919*
wnba_on_off_schema, *804*, *910*, *916*
wnba_pbp, 810
wnba_pbp(), *489*, *758*, *810*, *812*, *874*
wnba_pbps, 811
wnba_pbps(), *758*, *810–812*, *874*
wnba_playbyplayv3, 812
wnba_playbyplayv3(), *758*, *810–812*, *874*
wnba_player_crosswalk, 813
wnba_player_crosswalk(), *496*, *860*, *875*
wnba_player_dashboard_schema, *827–829*, *831, 832*
wnba_playerawards, 813
wnba_playerawards(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *815*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playercareerbycollege, 814
wnba_playercareerbycollege(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814*, *817*, *819*, *821*, *833*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playercareerbycollegerollup, 816
wnba_playercareerbycollegerollup(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *819*, *821*, *834–836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playercareerstats, 817
wnba_playercareerstats(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *821*, *834–836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playercareerstats_season_rankings_schema, *819*, *852*
wnba_playercareerstats_season_totals_schema, *818*, *851*
wnba_playercompare, 819
wnba_playercompare(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *834–836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playerdashboardbyclutch, 821
wnba_playerdashboardbyclutch(), *771*, *776*, *779*, *782*, *785*, *788*, *790*, *805, 806*, *814, 815*, *817*, *819*, *821*, *835, 836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*, *868, 869*, *873*, *885*
wnba_playerdashboardbygamesplits *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbygeneralsplits *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbylastngames *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbyopponent *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbyshootingsplits *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbyteamperformance *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerdashboardbyyearoveryear *(*wnba_playerdashboardbyclutch*)*, 821
wnba_playerestimatedmetrics, 834
wnba_playerestimatedmetrics(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834*, *836*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playerfantasyprofile, 835
wnba_playerfantasyprofile(), *763*, *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834*, *835*, *838, 839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playerfantasyprofilebargraph, 837
wnba_playerfantasyprofilebargraph(), *763*, *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *839*, *841*, *846–848*, *850*, *853*, *855*
wnba_playergamelog, 838
wnba_playergamelog(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838*, *841*, *846–848*, *850*, *853*, *855*
wnba_playergamelogs, 839
wnba_playergamelogs(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *846–848*,

*850*, *853*, *855*
wnba_playergamestreakfinder, 841
wnba_playergamestreakfinder(), *771*, *776*, *779*, *782*, *785*, *798*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *847, 848*, *850*, *853*, *855*, *905*
wnba_playerheadshot, 846
wnba_playerheadshot(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *846*, *848*, *850*, *853*, *855*
wnba_playerindex, 847
wnba_playerindex(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *846, 847*, *850*, *853*, *855*
wnba_playernextngames, 849
wnba_playernextngames(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *846–848*, *853*, *855*
wnba_playerprofilev2, 850
wnba_playerprofilev2(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *846–848*, *850*, *855*
wnba_players_on_court_schema, *811*
wnba_playervsplayer, 853
wnba_playervsplayer(), *771*, *776*, *779*, *782*, *785*, *805, 806*, *814, 815*, *817*, *819*, *821*, *834–836*, *838, 839*, *841*, *846–848*, *850*, *853*
wnba_possession_lineups, 856
wnba_possession_lineups(), *768*, *774*, *803*, *857*, *865*, *873*, *889*
wnba_rapm, 856
wnba_rapm(), *768*, *774*, *803*, *856*, *865*, *873*, *889*
wnba_referee_assignments, 857
wnba_schedule, 859
wnba_schedule(), *863, 864*, *874*, *919*
wnba_schedule_crosswalk, 860
wnba_schedule_crosswalk(), *496*, *813*, *875*
wnba_scoreboard(), *874*
wnba_scoreboard_east_conf_standings_by_day_schema, *862*
wnba_scoreboardv2, 860
wnba_scoreboardv2(), *859*, *864*, *874*, *919*
wnba_scoreboardv3, 863
wnba_scoreboardv3(), *859*, *863*, *874*, *919*
wnba_shooting_splits_assisted_by_schema, *832*, *884*
wnba_shooting_splits_schema, *831*, *883*
wnba_shot_zones, 864
wnba_shot_zones(), *768*, *774*, *803*, *856, 857*, *873*, *889*
wnba_shotchartdetail, 866
wnba_shotchartdetail(), *782*, *790*, *834*, *869*, *873*, *885*
wnba_shotchartdetail_league_averages_schema, *868, 869*, *873*
wnba_shotchartleaguewide, 868
wnba_shotchartleaguewide(), *774*, *776*, *779*, *782*, *785*, *788*, *790*, *793*, *798, 799*, *803*, *805–807*, *834*, *868*, *873*, *885*
wnba_shotchartlineupdetail, 869
wnba_shotchartlineupdetail(), *768*, *774*, *782*, *790*, *803*, *834*, *856, 857*, *865*, *868, 869*, *885*, *889*
wnba_stats, 873
wnba_team_crosswalk, 874
wnba_team_crosswalk(), *496*, *813*, *860*
wnba_team_dashboard_schema, *881–885*
wnba_team_summary_overall_schema, *887*, *910*
wnba_teamdashboardbyclutch, 875
wnba_teamdashboardbyclutch(), *782*, *788*, *790*, *793*, *834*, *868, 869*, *873*, *889*, *891–893*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_teamdashboardbygamesplits *(*wnba_teamdashboardbyclutch*)*, 875
wnba_teamdashboardbygeneralsplits *(*wnba_teamdashboardbyclutch*)*, 875
wnba_teamdashboardbylastngames *(*wnba_teamdashboardbyclutch*)*, 875
wnba_teamdashboardbyopponent *(*wnba_teamdashboardbyclutch*)*, 875
wnba_teamdashboardbyshootingsplits *(*wnba_teamdashboardbyclutch*)*,

875
wnba_teamdashboardbyteamperformance (wnba_teamdashboardbyclutch), 875
wnba_teamdashboardbyyearoveryear (wnba_teamdashboardbyclutch), 875
wnba_teamdashlineups, 886
wnba_teamdashlineups(), *768*, *774*, *793*, *803*, *856, 857*, *865*, *873*, *885*, *891–893*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_teamdetails, 889
wnba_teamdetails(), *793*, *885*, *889*, *892, 893*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_teamestimatedmetrics, 891
wnba_teamestimatedmetrics(), *793*, *885*, *889*, *891*, *893*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_teamgamelog, 892
wnba_teamgamelog(), *793*, *885*, *889*, *891, 892*, *895*, *905–907*, *911, 912*, *917, 918*
wnba_teamgamelogs, 893
wnba_teamgamelogs(), *793*, *885*, *889*, *891–893*, *905–907*, *911, 912*, *917, 918*
wnba_teamgamestreakfinder, 895
wnba_teamgamestreakfinder(), *793*, *798*, *846*, *885*, *889*, *891–893*, *895*, *906, 907*, *911, 912*, *917, 918*
wnba_teamhistoricalleaders, 905
wnba_teamhistoricalleaders(), *793*, *885*, *889*, *891–893*, *895*, *905*, *907*, *911, 912*, *917, 918*
wnba_teaminfocommon, 906
wnba_teaminfocommon(), *793*, *885*, *889*, *891–893*, *895*, *905, 906*, *911, 912*, *917, 918*
wnba_teamplayerdashboard (wnba_teamplayeronoffdetails), 907
wnba_teamplayeronoffdetails, 907
wnba_teamplayeronoffdetails(), *793*, *885*, *889*, *891–893*, *895*, *905–907*, *912*, *917, 918*
wnba_teamplayeronoffsummary (wnba_teamplayeronoffdetails), 907
wnba_teams, 911
wnba_teams(), *793*, *885*, *889*, *891–893*, *895*, *905–907*, *911*, *917, 918*
wnba_teamvsplayer, 912
wnba_teamvsplayer(), *793*, *885*, *889*, *891–893*, *895*, *905–907*, *911, 912*, *918*
wnba_teamyearbyyearstats, 918
wnba_teamyearbyyearstats(), *793*, *885*, *889*, *891–893*, *895*, *905–907*, *911, 912*, *917*
wnba_todays_scoreboard, 919
wnba_todays_scoreboard(), *810*, *859*, *863, 864*, *874*
wnba_videodetailsasset(), *920, 921*
wnba_videoevents, 920
wnba_videoevents(), *921*
wnba_videostatus, 921
wnba_videostatus(), *920*

year_to_season, 922