

Package ‘oddsapiR’

August 28, 2026

Title Access Live Sports Odds from the Odds API

Version 1.0.1

Description A utility to quickly obtain clean and tidy sports odds from The Odds API <<https://the-odds-api.com>>. Provides wrappers for every version 4 endpoint -- featured-market and single-event odds (including player props and alternate lines), historical odds snapshots, scores, events, participants, and usage-quota reporting -- returning tidy tibbles ready for analysis.

License MIT + file LICENSE

URL <https://oddsapiR.sportsdataverse.org/>,
<https://github.com/sportsdataverse/oddsapiR>

BugReports <https://github.com/sportsdataverse/oddsapiR/issues>

Depends R (>= 4.0.0)

Imports cli (>= 3.4.1), data.table (>= 1.14.0), dplyr (>= 1.0.10), glue, httr2 (>= 1.0.0), jsonlite, magrittr, rlang (>= 1.0.4), rvest (>= 1.0.0), tibble (>= 3.0), tidyr (>= 1.0.0)

Suggests crayon (>= 1.3.4), curl, DBI, ggplot2, ggrepel, gt, knitr, progressr (>= 0.6.0), Rcpp (>= 1.0.7), RcppParallel (>= 5.1.4), rmarkdown, RSQLite, stats, stringi, stringr (>= 1.5.0), testthat, tidyselect (>= 1.2.0), usethis (>= 1.6.0), xml2 (>= 1.3)

Encoding UTF-8

LazyData true

SystemRequirements pandoc (>= 1.12.3), pandoc-citeproc

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Saiem Gilani [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-7194-9067>>)

Maintainer Saiem Gilani <saiem.gilani@gmail.com>

Repository CRAN

Date/Publication 2026-08-27 23:20:02 UTC

Contents

csv_from_url	2
progressively	3
rds_from_url	3
register_toa	4
toa_event_markets	5
toa_event_odds	6
toa_event_odds_history	8
toa_quota	11
toa_requests	12
toa_sports	13
toa_sports_events	14
toa_sports_events_history	15
toa_sports_keys	17
toa_sports_odds	17
toa_sports_odds_history	19
toa_sports_participants	21
toa_sports_scores	22
Index	24

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

Value

a dataframe as created by data.table::fread()

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

<code>f</code>	a function to add progressr functionality to.
<code>p</code>	a progressor function as created by <code>progressr::progressor()</code>

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

<code>url</code>	a character url
------------------	-----------------

Value

a dataframe as created by [readRDS\(\)](#)

`register_toa`

Odds API Key Registration

Description

Save your API Key as a system environment variable `ODDS_API_KEY`

Usage

`toa_key()`

`has_toa_key()`

`check_toa_key()`

Details

To get access to an API key, follow the instructions at <https://the-odds-api.com>

Using the key:

You can save the key for consistent usage by adding `ODDS_API_KEY=XXXX-YOUR-API-KEY-HERE-XXXXX` to your `.Renviron` file (easily accessed via `usethis::edit_r_environ()`).

Run `usethis::edit_r_environ()`, a new script will pop open named `.Renviron`, **THEN** paste the following in the new script that pops up (**without** quotations)

```
ODDS_API_KEY = XXXX-YOUR-API-KEY-HERE-XXXXX
```

Save the script and restart your RStudio session, by clicking Session (in between Plots and Build) and click Restart R

(there also exists the shortcut `Ctrl + Shift + F10` to restart your session).

If set correctly, from then on you should be able to use any of the `toa_` functions without any other changes.

For less consistent usage:

At the beginning of every session or within an R environment, save your API key as the environment variable `ODDS_API_KEY` (**with** quotations) using a command like the following.

```
Sys.setenv(ODDS_API_KEY = "XXXX-YOUR-API-KEY-HERE-XXXXX")
```

Value

Called as a side-effect to ensure that a user has an API key stored in their environment before making a call to the Odds API service.

See Also

[toa_quota\(\)](#) to inspect your usage credits after any API call, [toa_requests\(\)](#) to explicitly query your remaining/used credits, and [toa_sports\(\)](#) to start exploring available sports. Part of the [SportsDataverse](#).

Other The Odds API: Account & Usage: [toa_quota\(\)](#), [toa_requests\(\)](#)

toa_event_markets	Find the available market keys for a single event through the Odds API
-------------------	---

Description

Get the available market keys for each bookmaker for a single event.

Returns the recently seen market keys per bookmaker for an event (no odds). This is not a comprehensive list of all supported markets – more keys are returned as bookmakers open more markets approaching commence time. Use the returned market_key values to request odds via [toa_event_odds\(\)](#).

Usage quota cost: 1 credit.

Usage

```
toa_event_markets(
  sport_key,
  event_id,
  regions = "us",
  bookmakers = NULL,
  date_format = "iso"
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up markets for. See toa_sports() for a full lookup of sport_key values.
event_id	(<i>string</i> , required): The event_id to look up markets for. See toa_sports_events() for a full lookup of event_id values.
regions	(<i>string</i> , required): The region(s) that determine which bookmakers appear. Multiple can be specified if comma delimited. Options include: • us • us2 • uk • eu • au
bookmakers	(<i>string</i> , optional): Comma-separated list of bookmakers to be returned. If both bookmakers and regions are specified, bookmakers takes precedence.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.

Value

A tibble with one row per bookmaker market available for the event:

col_name	types	description
id	character	Unique event id (echoes event_id).
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Event start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
bookmaker_key	character	Bookmaker slug, e.g. draftkings.
bookmaker	character	Bookmaker display title, e.g. DraftKings.
market_key	character	Available market key, e.g. player_points.
market_last_update	character	When this market was last seen for the bookmaker.

See Also

[toa_event_odds\(\)](#) to pull odds for the market keys discovered here, [toa_sports_odds\(\)](#) for featured-market odds across all events, and [toa_sports_events\(\)](#) to look up event_id values. Part of the [SportsDataverse](#).

Other The Odds API: Odds & Markets: [toa_event_odds\(\)](#), [toa_sports_odds\(\)](#)

Examples

```
try(toa_event_markets(sport_key = 'basketball_nba',
  event_id = '48db9c3293a52baab881d95d38f37a98',
  regions = 'us'))
```

toa_event_odds	Find odds for a single event, including player props and other markets
----------------	---

Description

Get bookmaker odds for a single event from The Odds API.

Unlike [toa_sports_odds\(\)](#) (featured markets only), this endpoint accepts any available market key, including player props and alternate lines. Look up an event_id with [toa_sports_events\(\)](#).

Usage quota cost: [number of unique markets returned] x [number of regions].

Usage

```
toa_event_odds(
  sport_key,
  event_id,
  regions = "us",
  markets = "spreads",
  odds_format = "decimal",
  date_format = "iso",
  bookmakers = NULL
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up odds for. See toa_sports() for a full lookup of sport_key values.
event_id	(<i>string</i> , required): The event_id to look up odds for. See toa_sports_events() for a full lookup of event_id values.
regions	(<i>string</i> , required): The region(s) to pull bookmakers from. Multiple can be specified if comma delimited. Options include: • us • us2 • uk • eu • au
markets	(<i>string</i> , optional): The markets to return. Multiple can be specified if comma delimited. In addition to h2h, spreads, totals and outrights, this endpoint accepts additional markets such as alternate lines and player props (e.g. player_points, player_pass_tds, player_shots_on_goal). See the betting markets documentation for the full list.
odds_format	(<i>string</i> , optional): Format in which to return odds. Options are decimal (default) or american.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.
bookmakers	(<i>string</i> , optional): Comma-separated list of bookmakers to be returned. If both bookmakers and regions are specified, bookmakers takes precedence. Bookmakers can be from any region. Every group of 10 bookmakers counts as 1 region against the usage quota.

Value

A long-format tibble with one row per bookmaker market outcome:

col_name	types	description
id	character	Unique event id (echoes event_id).
sport_key	character	Sport key, e.g. basketball_nba.

sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Game start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
bookmaker_key	character	Bookmaker slug, e.g. draftkings.
bookmaker	character	Bookmaker display title, e.g. DraftKings.
market_key	character	Market key, e.g. player_points.
market_last_update	character	When this market's odds were last updated.
outcomes_name	character	Outcome label (team, Over/Under, player name, etc.).
outcomes_description	character	Outcome description (player name for props); absent for featured markets.
outcomes_price	numeric	The price/odds for the outcome.
outcomes_point	numeric	The handicap/total/prop line, when applicable.

If the event exists but no bookmaker odds are posted yet for the requested markets/regions, a zero-row tibble with the same columns is returned.

See Also

[toa_sports_odds\(\)](#) for featured-market odds across all upcoming events, [toa_event_markets\(\)](#) to discover available market keys for an event, and [toa_sports_events\(\)](#) to look up event_id values. Part of the [SportsDataverse](#).

Other The Odds API: Odds & Markets: [toa_event_markets\(\)](#), [toa_sports_odds\(\)](#)

Examples

```
try(toa_event_odds(sport_key = 'basketball_nba',
  event_id = '48db9c3293a52baab881d95d38f37a98',
  regions = 'us',
  markets = 'player_points',
  odds_format = 'decimal',
  date_format = 'iso'))
```

toa_event_odds_history

Find historical odds for a single event through the Odds API

Description

Get bookmaker odds for a single event as they appeared at a point in time.

Like [toa_event_odds\(\)](#) but for a historical snapshot: accepts any available market key (including player props and alternate lines) and returns the odds at the snapshot closest to (and at or before) the requested date. Look up a historical event_id with [toa_sports_events_history\(\)](#). The

returned timestamp, previous_timestamp and next_timestamp columns let you page through snapshots.

Historical odds for additional markets are available from 2023-05-03 at 5-minute intervals; featured markets reach back to June 2020 on paid plans.

Usage quota cost: [number of unique markets returned] x [number of regions] (the standard rate, not the 10x historical multiplier). Snapshots with no data do not count against the quota.

Usage

```
toa_event_odds_history(  
    sport_key,  
    event_id,  
    date,  
    regions = "us",  
    markets = "h2h",  
    odds_format = "decimal",  
    date_format = "iso",  
    bookmakers = NULL  
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up odds for. See toa_sports() for a full lookup of sport_key values.
event_id	(<i>string</i> , required): The event_id to look up odds for. See toa_sports_events_history() for a full lookup of event_id values.
date	(<i>string</i> , required): The timestamp of the data snapshot to return, in ISO 8601 format (e.g. 2023-10-10T12:15:00Z). The API returns the closest snapshot equal to or earlier than this value.
regions	(<i>string</i> , required): The region(s) to pull bookmakers from. Multiple can be specified if comma delimited. Options include: • us • us2 • uk • eu • au
markets	(<i>string</i> , optional): The markets to return. Multiple can be specified if comma delimited. In addition to h2h, spreads, totals and outrights, this endpoint accepts additional markets such as alternate lines and player props. See the betting markets documentation .
odds_format	(<i>string</i> , optional): Format in which to return odds. Options are decimal (default) or american.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.

bookmakers (*string*, optional): Comma-separated list of bookmakers to be returned. If both bookmakers and regions are specified, bookmakers takes precedence. Every group of 10 bookmakers counts as 1 region against the usage quota.

Value

A long-format tibble with one row per bookmaker market outcome:

col_name	types	description
timestamp	character	Snapshot timestamp returned (closest at/before date).
previous_timestamp	character	Preceding available snapshot; use as date to page back.
next_timestamp	character	Next available snapshot; use as date to page forward.
id	character	Unique event id (echoes event_id).
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Game start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
bookmaker_key	character	Bookmaker slug, e.g. draftkings.
bookmaker	character	Bookmaker display title, e.g. DraftKings.
bookmaker_last_update	character	When this bookmaker's odds were last updated.
market_key	character	Market key, e.g. player_points.
market_last_update	character	When this market's odds were last updated.
outcomes_name	character	Outcome label (team, Over/Under, player name, etc.).
outcomes_description	character	Outcome description (player name for props); absent for featured markets.
outcomes_price	numeric	The price/odds for the outcome.
outcomes_point	numeric	The handicap/total/prop line, when applicable.

If the event exists at the requested snapshot but no bookmaker odds were posted for the requested markets/regions, a zero-row tibble with the same columns is returned.

See Also

[toa_sports_events_history\(\)](#) to look up historical event_id values, [toa_sports_odds_history\(\)](#) for featured-market odds across all events at a snapshot, and [toa_event_odds\(\)](#) for current event odds. Part of the [SportsDataverse](#).

Other The Odds API: Historical: [toa_sports_events_history\(\)](#), [toa_sports_odds_history\(\)](#)

Examples

```
try(toa_event_odds_history(sport_key = 'basketball_nba',
                           event_id = '93af4b300a4c0dded909234ea32e9abd',
                           date = '2024-01-15T12:15:00Z',
                           regions = 'us',
                           markets = 'h2h',
                           odds_format = 'decimal',
                           date_format = 'iso'))
```

toa_quota

Inspect your Odds API usage quota

Description

Returns the usage-quota headers reported by The Odds API on the **most recent** `toa_*`() call made in this R session. The Odds API returns three headers with every response:

- `x-requests-remaining` – usage credits remaining until the quota resets
- `x-requests-used` – usage credits used since the last quota reset
- `x-requests-last` – the usage cost of the last API call

These same values are attached as attributes (`oddsapiR_requests_remaining`, `oddsapiR_requests_used`, `oddsapiR_requests_last`) to every tibble returned by a `toa_*`() function, and are echoed when the tibble is printed.

Usage

```
toa_quota()
```

Value

A one-row tibble with columns `requests_remaining`, `requests_used` and `requests_last`, or `NULL` if no API call has been made yet this session.

See Also

[toa_requests\(\)](#) to explicitly query usage via an API round-trip, [toa_key\(\)](#) / [register_toa](#) to set up your API key, and [toa_sports\(\)](#) for your first data call. Part of the [SportsDataaverse](#).

Other The Odds API: Account & Usage: [register_toa](#), [toa_requests\(\)](#)

Examples

```
try(toa_sports())
try(toa_quota())
```

toa_requests	Find out your usage and remaining calls for your key from The Odds API
--------------	---

Description

Get your usage and remaining calls for your key from The Odds API.

The values are read from the x-requests-remaining and x-requests-used response headers returned with every API call. This check is performed against the free /v4/sports endpoint, so it does not consume any quota.

Usage quota cost: Free.

Usage

```
toa_requests()
```

Value

A tibble of The Odds API key usage with the following columns:

col_name	types	description
requests_remaining	integer	Usage credits remaining until the monthly quota resets.
requests_used	integer	Usage credits consumed since the last quota reset.

See Also

[toa_quota\(\)](#) to read cached usage from the most recent call without a network round-trip, [toa_key\(\)](#) / [register_toa](#) to configure your API key, and [toa_sports\(\)](#) to start pulling data. Part of the [Sports-Dataverse](#).

Other The Odds API: Account & Usage: [register_toa](#), [toa_quota\(\)](#)

Examples

```
try(toa_requests())
```

toa_sports

Find sports for which odds are accessible through the Odds API**Description****Get the sports for which The Odds API provides coverage.**

Returns a list of in-season sports. The key returned here is the `sport_key` consumed by every other `toa_*`() endpoint.

Usage quota cost: Free. This endpoint does not count against your quota.

Usage

```
toa_sports(all_sports = TRUE)
```

Arguments

`all_sports` (*Logical*, optional): If TRUE, returns all sports (including out-of-season). If FALSE, returns only in-season sports. Defaults to TRUE.

Value

A tibble of the sports for which The Odds API provides coverage:

col_name	types	description
key	character	Sport key, e.g. americanfootball_nfl. Use as <code>sport_key</code> elsewhere.
group	character	Sport group / category, e.g. American Football.
title	character	Human-readable sport title, e.g. NFL.
description	character	Description of the sport or competition.
active	logical	TRUE if the sport is currently in season.
has_outrights	logical	TRUE if the sport offers outright (futures) markets.

See Also

[toa_sports_events\(\)](#) to list upcoming events for a sport, [toa_sports_scores\(\)](#) for live/recent scores, and [toa_sports_odds\(\)](#) to pull featured-market odds. Part of the [SportsDataaverse](#).

Other The Odds API: Sports & Events: [toa_sports_events\(\)](#), [toa_sports_participants\(\)](#), [toa_sports_scores\(\)](#)

Examples

```
try(toa_sports(all_sports = TRUE))
```

toa_sports_events	Find in-play and upcoming events for a sport through the Odds API
-------------------	--

Description

Get the list of in-play and pre-match events for a sport.

Returns a list of events without odds. The id returned here is the event_id consumed by [toa_event_odds\(\)](#) and [toa_event_markets\(\)](#).

Usage quota cost: Free. This endpoint does not count against your quota.

Usage

```
toa_sports_events(
  sport_key,
  date_format = "iso",
  event_ids = NULL,
  commence_time_from = NULL,
  commence_time_to = NULL,
  include_rotation_numbers = FALSE
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up events for. See toa_sports() for a full lookup of sport_key values.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.
event_ids	(<i>string</i> , optional): Comma-separated list of event ids to filter the response to. Defaults to NULL (all events).
commence_time_from	(<i>string</i> , optional): Filter to events that commence on and after this ISO 8601 timestamp (e.g. 2023-09-09T00:00:00Z).
commence_time_to	(<i>string</i> , optional): Filter to events that commence on and before this ISO 8601 timestamp (e.g. 2023-09-09T00:00:00Z).
include_rotation_numbers	(<i>logical</i> , optional): If TRUE, include the home_rotation and away_rotation columns when available. Defaults to FALSE.

Value

A tibble with one row per event:

col_name	types	description
----------	-------	-------------

id	character	Unique event id. Use as event_id elsewhere.
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Event start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
home_rotation	integer	Home rotation number (only when include_rotation_numbers = TRUE).
away_rotation	integer	Away rotation number (only when include_rotation_numbers = TRUE).

See Also

[toa_sports\(\)](#) to discover sport_key values, [toa_event_odds\(\)](#) to pull odds for a specific event by its id, and [toa_event_markets\(\)](#) to see available market keys per bookmaker. Part of the [SportsDataverse](#).

Other The Odds API: Sports & Events: [toa_sports\(\)](#), [toa_sports_participants\(\)](#), [toa_sports_scores\(\)](#)

Examples

```
try(toa_sports_events(sport_key = 'basketball_nba'))
```

toa_sports_events_history

Find historical events for a sport through the Odds API

Description

Get the list of events for a sport as they appeared at a point in time.

Returns the events (without odds) that existed at the snapshot closest to (and at or before) the requested date. The returned timestamp, previous_timestamp and next_timestamp columns let you page through snapshots. The id returned here is the event_id consumed by [toa_event_odds_history\(\)](#).

Historical data is available on paid usage plans from June 2020.

Usage quota cost: 1 credit (0 if no events are found at the snapshot).

Usage

```
toa_sports_events_history(
  sport_key,
  date,
  date_format = "iso",
  event_ids = NULL,
  commence_time_from = NULL,
  commence_time_to = NULL,
  include_rotation_numbers = FALSE
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up events for. See toa_sports() for a full lookup of sport_key values.
date	(<i>string</i> , required): The timestamp of the data snapshot to return, in ISO 8601 format (e.g. 2023-10-10T12:15:00Z). The API returns the closest snapshot equal to or earlier than this value.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.
event_ids	(<i>string</i> , optional): Comma-separated list of event ids to filter the response to. Defaults to NULL (all events).
commence_time_from	(<i>string</i> , optional): Filter to events that commence on and after this ISO 8601 timestamp.
commence_time_to	(<i>string</i> , optional): Filter to events that commence on and before this ISO 8601 timestamp.
include_rotation_numbers	(<i>logical</i> , optional): If TRUE, include the home_rotation and away_rotation columns when available. Defaults to FALSE.

Value

A tibble with one row per historical event:

col_name	types	description
timestamp	character	Snapshot timestamp returned (closest at/before date).
previous_timestamp	character	Preceding available snapshot; use as date to page back.
next_timestamp	character	Next available snapshot; use as date to page forward.
id	character	Unique event id. Use as event_id elsewhere.
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Event start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
home_rotation	integer	Home rotation number (only when include_rotation_numbers = TRUE).
away_rotation	integer	Away rotation number (only when include_rotation_numbers = TRUE).

See Also

[toa_event_odds_history\(\)](#) to pull odds for a historical event by its id, [toa_sports_odds_history\(\)](#) for featured-market odds at a historical snapshot, and [toa_sports_events\(\)](#) for current events. Part of the [SportsDataverse](#).

Other The Odds API: Historical: [toa_event_odds_history\(\)](#), [toa_sports_odds_history\(\)](#)

Examples

```
try(toa_sports_events_history(sport_key = 'basketball_nba',
                             date = '2024-01-15T12:15:00Z'))
```

toa_sports_keys	Sports for which odds are accessible through the Odds API
-----------------	---

Description

A data set mapping Sports Events/League names to keys for end-user ease.

Usage

```
data(toa_sports_keys)
```

Format

A data frame with 5 variables:
key - Sport key group - Sport group (non-league description) title - Sport title description - Sport description has_outrights - Whether the sport or event has outright victories.

toa_sports_odds	Find odds for the sports which are accessible through the Odds API
-----------------	--

Description

Get the featured-market odds for a sport which The Odds API covers.
Returns a list of upcoming and live games with featured bookmaker odds for the specified markets and regions. For player props and other additional markets, use [toa_event_odds\(\)](#) instead.
Usage quota cost: [number of markets] x [number of regions].

Usage

```
toa_sports_odds(  
  sport_key,  
  regions = "us",  
  markets = "spreads",  
  odds_format = "decimal",  
  date_format = "iso"  
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up odds for. See toa_sports() for a full lookup of sport_key values.
regions	(<i>string</i> , required): The region(s) to pull bookmakers from. Multiple can be specified if comma delimited. Options include: • us • us2 • uk • eu • au
markets	(<i>string</i> , optional): The featured markets to return. Multiple can be specified if comma delimited. Options include: • h2h (moneyline) • spreads (points handicaps) • totals (over/under) • outrights (futures, select sports only)
odds_format	(<i>string</i> , optional): Format in which to return odds. Options are decimal (default) or american.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.

Value

A long-format tibble with one row per bookmaker market outcome:

col_name	types	description
id	character	Unique event id.
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Game start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
bookmaker_key	character	Bookmaker slug, e.g. draftkings.
bookmaker	character	Bookmaker display title, e.g. DraftKings.
bookmaker_last_update	character	When this bookmaker's odds were last updated.
market_key	character	Market key, e.g. spreads.
market_last_update	character	When this market's odds were last updated.
outcomes_name	character	Outcome label (team name, Over/Under, etc.).
outcomes_price	numeric	The price/odds for the outcome.
outcomes_point	numeric	The handicap/total line, when applicable.

See Also

[toa_event_odds\(\)](#) for a single event's odds (including player props and alternate lines), [toa_event_markets\(\)](#) to list a game's available market keys, and [toa_sports\(\)](#) to find sport_key values. Part of the [SportsDataverse](#).

Other The Odds API: Odds & Markets: [toa_event_markets\(\)](#), [toa_event_odds\(\)](#)

Examples

```
try(toa_sports_odds(sport_key = 'basketball_nba',  
                    regions = 'us',  
                    markets = 'spreads',  
                    odds_format = 'decimal',  
                    date_format = 'iso'))
```

toa_sports_odds_history

Find historical featured-market odds through the Odds API

Description

Get historical featured-market odds for a sport which The Odds API covers.

Returns the list of events and featured bookmaker odds (h2h, spreads, totals) as they appeared at the snapshot closest to (and at or before) the requested date. The response is wrapped in a snapshot envelope; the returned timestamp, previous_timestamp and next_timestamp columns let you page backward/forward in time. This endpoint was previously `/v4/sports/{sport}/odds-history`, which has been deprecated in favour of `/v4/historical/sports/{sport}/odds`.

Historical data is available on paid usage plans from June 2020.

Usage quota cost: 10 x [number of markets] x [number of regions]. Snapshots with no events do not count against the quota.

Usage

```
toa_sports_odds_history(  
  sport_key,  
  date,  
  regions = "us",  
  markets = "spreads",  
  odds_format = "decimal",  
  date_format = "iso",  
  event_ids = NULL,  
  bookmakers = NULL  
)
```

Arguments

sport_key	(<i>string</i> , required): The sport_key to look up odds for. See toa_sports() for a full lookup of sport_key values.
-----------	---

date	(<i>string</i> , required): The timestamp of the data snapshot to return, in ISO 8601 format (e.g. 2023-03-18T12:15:00Z). The API returns the closest snapshot equal to or earlier than this value.
regions	(<i>string</i> , required): The region(s) to pull bookmakers from. Multiple can be specified if comma delimited. Options include: • us • us2 • uk • eu • au
markets	(<i>string</i> , optional): The featured markets to return. Multiple can be specified if comma delimited. Options include h2h, spreads, totals.
odds_format	(<i>string</i> , optional): Format in which to return odds. Options are decimal (default) or american.
date_format	(<i>string</i> , optional): Format of returned timestamps. Options are iso (default) or unix.
event_ids	(<i>string</i> , optional): Comma-separated list of event ids to filter the snapshot to. Defaults to NULL (all events).
bookmakers	(<i>string</i> , optional): Comma-separated list of bookmakers to be returned. If both bookmakers and regions are specified, bookmakers takes precedence. Every group of 10 bookmakers counts as 1 region against the usage quota.

Value

A long-format tibble with one row per bookmaker market outcome:

col_name	types	description
timestamp	character	Snapshot timestamp returned (closest at/before date).
previous_timestamp	character	Preceding available snapshot; use as date to page back.
next_timestamp	character	Next available snapshot; use as date to page forward.
id	character	Unique event id.
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Game start time (ISO 8601 string or unix seconds).
home_team	character	Home team name.
away_team	character	Away team name.
bookmaker_key	character	Bookmaker slug, e.g. draftkings.
bookmaker	character	Bookmaker display title, e.g. DraftKings.
bookmaker_last_update	character	When this bookmaker's odds were last updated.
market_key	character	Market key, e.g. spreads.
market_last_update	character	When this market's odds were last updated.
outcomes_name	character	Outcome label (team name, Over/Under, etc.).
outcomes_price	numeric	The price/odds for the outcome.
outcomes_point	numeric	The handicap/total line (spreads/totals only).

See Also

[toa_event_odds_history\(\)](#) for historical odds on a single event (including player props), [toa_sports_events_history\(\)](#) to list events at a historical snapshot, and [toa_sports_odds\(\)](#) for current featured-market odds. Part of the [SportsDataverse](#).

Other The Odds API: Historical: [toa_event_odds_history\(\)](#), [toa_sports_events_history\(\)](#)

Examples

```
try(toa_sports_odds_history(sport_key = 'basketball_nba',
                           date = '2024-01-15T12:15:00Z',
                           regions = 'us',
                           markets = 'spreads',
                           odds_format = 'decimal',
                           date_format = 'iso'))
```

toa_sports_participants

Find the participants (teams or players) for a sport through the Odds API

Description

Get the list of participants for a sport.

Depending on the sport a participant is either a team (e.g. NBA) or an individual (e.g. tennis). The list is a whitelist used to define events and may include participants that are no longer active. This endpoint does not return individual players on a team.

Usage quota cost: 1 credit.

Usage

```
toa_sports_participants(sport_key)
```

Arguments

sport_key (string, required): The sport_key to look up participants for. See [toa_sports\(\)](#) for a full lookup of sport_key values.

Value

A tibble with one row per participant:

col_name	types	description
sport_key	character	Sport key the participants belong to (echoes the sport_key argument).
id	character	Unique participant id, e.g. par_01hqmkq6fdf1pvq7jgdd7hdmpf.

full_name character Participant name – a team name or player name depending on the sport.

See Also

[toa_sports\(\)](#) to discover sport_key values, [toa_sports_events\(\)](#) to list upcoming events, and [toa_sports_scores\(\)](#) for live/recent scores. Part of the [SportsDataverse](#).

Other The Odds API: Sports & Events: [toa_sports\(\)](#), [toa_sports_events\(\)](#), [toa_sports_scores\(\)](#)

Examples

```
try(toa_sports_participants(sport_key = 'basketball_nba'))
```

toa_sports_scores	Find scores for the sports which are accessible through the Odds API
-------------------	---

Description

Get the scores for the sports which The Odds API provides coverage.

Returns live and recently completed games for a sport, including scores for games completed within the last 3 days. Live scores update roughly every 30 seconds.

Usage quota cost: 1 credit when days_from is not supplied; 2 credits when days_from is supplied.

Usage

```
toa_sports_scores(sport_key, days_from = NULL, date_format = "iso")
```

Arguments

- | | |
|-------------|--|
| sport_key | (<i>string</i> , required): The sport_key to look up scores for. See toa_sports() for a full lookup of sport_key values. |
| days_from | (<i>integer</i> , optional): Number of days in the past from which to return completed games, an integer from 1 to 3. If NULL (default), only live and upcoming games are returned. |
| date_format | (<i>string</i> , optional): Format of returned timestamps. Options are: <ul style="list-style-type: none"> • iso (ISO 8601, the default) • unix (epoch seconds) |

Value

A tibble of scores for the sport The Odds API provides coverage for:

col_name	types	description
id	character	Unique event id.
sport_key	character	Sport key, e.g. basketball_nba.
sport_title	character	Human-readable sport title, e.g. NBA.
commence_time	character	Game start time (ISO 8601 string or unix seconds).
completed	logical	TRUE if the game has finished.
home_team	character	Home team name.
away_team	character	Away team name.
scores	list	Per-team name/score pairs; NULL/NA before scores post.
last_update	character	Time the scores were last updated; NA until a game is live.

See Also

[toa_sports\(\)](#) to discover sport_key values, [toa_sports_events\(\)](#) to list upcoming events, and [toa_sports_odds\(\)](#) for live/upcoming betting lines. Part of the [SportsDataaverse](#).

Other The Odds API: Sports & Events: [toa_sports\(\)](#), [toa_sports_events\(\)](#), [toa_sports_participants\(\)](#)

Examples

```
try(toa_sports_scores(sport_key = 'basketball_nba',
                     days_from = NULL,
                     date_format = 'iso'))
```

Index

- * **Betting**
 - toa_event_odds, [6](#)
 - toa_event_odds_history, [8](#)
 - toa_sports_odds, [17](#)
 - toa_sports_odds_history, [19](#)
- * **Events**
 - toa_sports_events, [14](#)
 - toa_sports_events_history, [15](#)
- * **Internal**
 - csv_from_url, [2](#)
 - progressively, [3](#)
 - rds_from_url, [3](#)
- * **Lines**
 - toa_event_odds, [6](#)
 - toa_event_odds_history, [8](#)
 - toa_sports_odds, [17](#)
 - toa_sports_odds_history, [19](#)
- * **Markets**
 - toa_event_markets, [5](#)
- * **Participants**
 - toa_sports_participants, [21](#)
- * **Scores**
 - toa_sports_scores, [22](#)
- * **Sports**
 - toa_sports, [13](#)
- * **The Odds API: Account & Usage**
 - register_toa, [4](#)
 - toa_quota, [11](#)
 - toa_requests, [12](#)
- * **The Odds API: Historical**
 - toa_event_odds_history, [8](#)
 - toa_sports_events_history, [15](#)
 - toa_sports_odds_history, [19](#)
- * **The Odds API: Odds & Markets**
 - toa_event_markets, [5](#)
 - toa_event_odds, [6](#)
 - toa_sports_odds, [17](#)
- * **The Odds API: Sports & Events**
 - toa_sports, [13](#)
 - toa_sports_events, [14](#)
 - toa_sports_participants, [21](#)
 - toa_sports_scores, [22](#)
- * **Usage**
 - toa_requests, [12](#)
- * **datasets**
 - toa_sports_keys, [17](#)
- check_toa_key (register_toa), [4](#)
- csv_from_url, [2](#)
- data.table::fread(), [2](#)
- has_toa_key (register_toa), [4](#)
- progressively, [3](#)
- rds_from_url, [3](#)
- readRDS(), [3](#)
- register_toa, [4](#), [11](#), [12](#)
- toa_event_markets, [5](#)
- toa_event_markets(), [8](#), [14](#), [15](#), [18](#), [19](#)
- toa_event_odds, [6](#)
- toa_event_odds(), [5](#), [6](#), [8](#), [10](#), [14](#), [15](#), [17–19](#)
- toa_event_odds_history, [8](#)
- toa_event_odds_history(), [15](#), [16](#), [21](#)
- toa_key (register_toa), [4](#)
- toa_key(), [11](#), [12](#)
- toa_quota, [11](#)
- toa_quota(), [5](#), [12](#)
- toa_requests, [12](#)
- toa_requests(), [5](#), [11](#)
- toa_sports, [13](#)
- toa_sports(), [5](#), [7](#), [9](#), [11](#), [12](#), [14–16](#), [18](#), [19](#), [21–23](#)
- toa_sports_events, [14](#)
- toa_sports_events(), [5–8](#), [13](#), [16](#), [22](#), [23](#)
- toa_sports_events_history, [15](#)
- toa_sports_events_history(), [8–10](#), [21](#)
- toa_sports_keys, [17](#)

toa_sports_odds, [17](#)
toa_sports_odds(), [6](#), [8](#), [13](#), [21](#), [23](#)
toa_sports_odds_history, [19](#)
toa_sports_odds_history(), [10](#), [16](#)
toa_sports_participants, [21](#)
toa_sports_participants(), [13](#), [15](#), [23](#)
toa_sports_scores, [22](#)
toa_sports_scores(), [13](#), [15](#), [22](#)