

Package ‘oddsrelay’

October 21, 2026

Title Client for the 'OddsRelay' Sports Odds API

Version 0.1.0

Description Provides one function for each read endpoint of the 'OddsRelay' web service <<https://oddsrelay.io/docs>>, which serves bookmaker and exchange odds for the UK and Ireland. Requests ask for compressed replies, send entity tags so that unchanged data is not transferred again, wait as the service asks when it is busy, and signal refusals as classed conditions that carry the service's error code and request id. Replies are returned as lists and data frames, with the token and rate limit headers attached as attributes.

License MIT + file LICENSE

URL <https://github.com/oddsrelay/oddsrelay-r>,
<https://oddsrelay.io/docs>

BugReports <https://github.com/oddsrelay/oddsrelay-r/issues>

Depends R (>= 4.1.0)

Imports htr2 (>= 1.2.0), jsonlite, stats, utils

Suggests testthat (>= 3.1.7), withr

Config/oddsrelay/contract-version 2026-07-08

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author James OddsRelay [aut, cre],
OddsRelay [cph]

Maintainer James OddsRelay <hello@oddsrelay.io>

Repository CRAN

Date/Publication 2026-10-21 14:40:02 UTC

Contents

or_odds	2
or_sports	3
or_status	4
or_usage	5
Index	6

or_odds	<i>Matched boards and raw odds</i>
---------	------------------------------------

Description

or_odds() reads one matched board, where each selection pairs bookmaker back prices with exchange lay prices. or_raw() reads bookmaker and exchange prices by event, unpaired, a page at a time. or_event() reads one event’s raw prices. These calls spend tokens: pass quote = TRUE to learn what a call costs without making it.

Usage

```
or_odds(type, ..., .key = NULL)

or_raw(..., .key = NULL)

or_event(id, ..., .key = NULL)
```

Arguments

type	The board: "standard", "2up", "dutching", "each-way", "extra-place" or "bog".
...	Filters, named as in the API reference, such as region, sports, bookmakers or markets. snake_case names work too (odds_format is sent as oddsFormat). A vector is sent as one comma-separated value, TRUE as true, and a date-time as a UTC timestamp. The API refuses names and values it does not know.
.key	Your API key. Left out, it is read from the ODDSRELAY_KEY environment variable.
id	An event id: event_id on a board, or id on a raw page or an or_events() row.

Value

The decoded JSON body as a list, with arrays of records as data frames; a format = "bare" call returns the data frame alone. The reply’s token and rate limit headers ride along as attributes: see [oddsrelay-package](#).

Examples

```
# What would it cost?
or_odds("standard", region = "uk", sports = "soccer", quote = TRUE)

board <- or_odds("standard", region = "uk", sports = "soccer", markets = "h2h")
board$data[, c("commence_time", "home_team", "away_team")]
attr(board, "tokens")

page <- or_raw(region = "uk", sports = "tennis", limit = 50)
page$meta$next_cursor
```

or_sports	<i>Sports, events, bookmakers and regions</i>
-----------	---

Description

The keys the odds calls filter on. `or_sports()` lists sport and competition keys and `or_events()` the upcoming events for one sport key, both a page at a time (pass `meta$next_cursor` back as `cursor`). `or_bookmakers()` lists the bookmakers and exchanges with their keys, and `or_regions()` the region codes open to your key. These calls spend no tokens.

Usage

```
or_sports(..., .key = NULL)

or_events(sport, ..., .key = NULL)

or_bookmakers(..., .key = NULL)

or_regions(.key = NULL)
```

Arguments

<code>...</code>	Filters, named as in the API reference, such as <code>region</code> , <code>sports</code> , <code>bookmakers</code> or <code>markets</code> . <code>snake_case</code> names work too (<code>odds_format</code> is sent as <code>oddsFormat</code>). A vector is sent as one comma-separated value, <code>TRUE</code> as <code>true</code> , and a date-time as a UTC timestamp. The API refuses names and values it does not know.
<code>.key</code>	Your API key. Left out, it is read from the <code>ODDSRELAY_KEY</code> environment variable.
<code>sport</code>	A sport key from <code>or_sports()</code> .

Value

The decoded JSON body as a list, with arrays of records as data frames; a `format = "bare"` call returns the data frame alone. The reply's token and rate limit headers ride along as attributes: see [oddsrelay-package](#).

Examples

```
sports <- or_sports(region = "uk")
head(sports$data[, c("key", "title")])

or_events("tennis", region = "uk")
or_bookmakers(format = "bare")
or_regions()
```

or_status	<i>Service state</i>
-----------	----------------------

Description

or_status() says whether each product is serving and needs no key. or_coverage() lists which bookmakers, exchanges and sports each product carries; it needs no key, and gives more detail with one. or_health() says when each board was last processed and whether any is behind. or_healthz() is a plain-text liveness check and or_openapi() returns the API description; neither needs a key.

Usage

```
or_status(..., .key = NULL)

or_coverage(..., .key = NULL)

or_health(.key = NULL)

or_healthz()

or_openapi()
```

Arguments

...	Filters, named as in the API reference, such as region, sports, bookmakers or markets. snake_case names work too (odds_format is sent as oddsFormat). A vector is sent as one comma-separated value, TRUE as true, and a date-time as a UTC timestamp. The API refuses names and values it does not know.
.key	Your API key. Left out, it is read from the ODDSRELAY_KEY environment variable.

Value

As for [or_odds\(\)](#). or_healthz() returns a single string and or_openapi() a list that keeps the description's own structure.

Examples

```
or_status(region = "uk")
names(or_coverage(region = "uk")$products)
cat(or_healthz())
or_openapi()$info$version
```

or_usage	<i>Your account</i>
----------	---------------------

Description

or_usage() returns the tokens your account has used and has left, and what your key may read.
or_pricing() returns the token rate card. Neither spends tokens.

Usage

```
or_usage(.key = NULL)

or_pricing(.key = NULL)
```

Arguments

.key	Your API key. Left out, it is read from the ODDSRELAY_KEY environment variable.
------	---

Value

The decoded JSON body as a list, with arrays of records as data frames; a format = "bare" call returns the data frame alone. The reply's token and rate limit headers ride along as attributes: see [oddsrelay-package](#).

Examples

```
or_usage()$tokens
or_pricing()
```

Index

oddsrelay-package, [2](#), [3](#), [5](#)
or_bookmakers (or_sports), [3](#)
or_coverage (or_status), [4](#)
or_event (or_odds), [2](#)
or_events (or_sports), [3](#)
or_health (or_status), [4](#)
or_healthz (or_status), [4](#)
or_odds, [2](#)
or_odds(), [4](#)
or_openapi (or_status), [4](#)
or_pricing (or_usage), [5](#)
or_raw (or_odds), [2](#)
or_regions (or_sports), [3](#)
or_sports, [3](#)
or_status, [4](#)
or_usage, [5](#)