

# Package ‘opendotaR’

October 14, 2022

**Type** Package

**Title** Interface for OpenDota API

**Version** 0.1.4

**Author** Kari Gunnarsson

**Maintainer** Kari Gunnarsson <kari.gunnarsson@outlook.com>

**Description** Enables the usage of the OpenDota API from <<https://www.opendota.com/>>, get game lists, and download JSON's of parsed replays from the OpenDota API. Also has functionality to execute own code to extract the specific parts of the JSON file.

**License** GPL

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Imports** jsonlite, dplyr, lubridate

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-09-11 07:39:36 UTC

## R topics documented:

api_delay . . . . .	2
get_games . . . . .	2
get_game_list . . . . .	3
get_latest_games . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

api_delay	<i>API Delay function</i>
-----------	---------------------------

---

**Description**

Function that controls the delay between API calls to opendota, user can specify the wait\_time, but opendota asks you to limit yourself to 1 call per second

**Usage**

```
api_delay(start_time, wait_time = 1)
```

**Arguments**

start_time	Time of last API call
wait_time	Desired wait time between API calls

**Value**

There is no return, it simply sleeps the system for whatever time needed to reach wait\_time

**Examples**

```
## Not run:  
api_delay(start_time, wait_time)  
  
## End(Not run)
```

---

get_games	<i>Fetch the games from the opendota API.</i>
-----------	---

---

**Description**

Takes a vector of numerical value match ID's of dota2 replays, and attempts to fetch them from the opendota API only parsed matches are output.

**Usage**

```
get_games(game_vec, wait_time = 1, output = "all", verbose = TRUE)
```

**Arguments**

game_vec	Numeric vector of match ID's
wait_time	how long to wait (in seconds) between each API call, default is 1 sec (opendota asks you not to send more than 1 call per second)
output	Defaulted to "all", which will extract entire JSON, if not all, it should have the path to an R file that will be sourced and create some output, not the R file must also output to output_list()
verbose	Give live information on status of parsing, if FALSE no text is output to console.

**Value**

Returns a list of objects, if output == "all" it's a list of JSON outputs.

**Examples**

```
## Not run:
match_ids <- get_game_list(num_matches = 100,
  from_time = "20170101",
  to_time = "20170423",
  min_mmr = 4000)
get_games(match_ids)

## End(Not run)
```

---

get_game_list	<i>Get list of games / Match ID's</i>
---------------	---------------------------------------

---

**Description**

Create an SQL query to opendotas API and extracts a list of games from the public\_matches table. This is only a sample of matches, not all are included here. Returns a vector of match ID's ready for use in the get\_games() function.

**Usage**

```
get_game_list(num_matches, from_time, to_time, min_mmr = 1,
  min_duration = 1200, num_open_profile = 0)
```

**Arguments**

num_matches	Number of matches you want to extract
from_time	Earliest time of match in YMD text format.
to_time	Latest start time of the match in YMD text format.
min_mmr	Minimum average MMR of the match (defaulted to 1)
min_duration	Minium match duration in seconds, defaulted to 1200 (20 minutes)
num_open_profile	Minium number of open profiles in the game. Higher number here gives higher percentage of games that are actually parsed.

**Value**

Returns data frame of results fulfilling the parameters input.

**Examples**

```
## Not run:
match_ids <- get_game_list(num_matches = 100,
  from_time = "20170101" ,
  to_time = "20170423",
  min_mmr = 4000)

## End(Not run)
```

---

get_latest_games	<i>Obtain the latest parsed games, this is a good function to use if you're not picky on which dates or MMR your data is, but want fast data. The latest games will always have parsed games opposed to the general game list gotten from get_game_list(), wich only contains 5 - 10 games.</i>
------------------	---

---

**Description**

Obtain the latest parsed games, this is a good function to use if you're not picky on which dates or MMR your data is, but want fast data. The latest games will always have parsed games opposed to the general game list gotten from get\_game\_list(), wich only contains 5 - 10 games.

**Usage**

```
get_latest_games(num_games, min_duration = 1200, wait_time = 1,
  output = "all")
```

**Arguments**

num_games	Min number of games you want to obtain (could get 1-10 more)
min_duration	Do you want to exclude games below a certain duration threshold? We default it to 1200seconds (20 minutes), as super short games often contain early abandons and griefers.
wait_time	Wait time between API calls, default to 1.00 (which is what opendota wants you to stay below, so don't change unless you have a good reason and talked to opendota about it).
output	Defaulted to "all", which will extract entire JSON, if not all, it should have the path to an R file that will be sourced and create some output, note the R file must also output to output_list()

*get\_latest\_games*

5

### **Examples**

```
## Not run:  
parsed_games <- get_latest_games(100)  
  
## End(Not run)
```

# Index

`api_delay`, 2

`get_game_list`, 3

`get_games`, 2

`get_latest_games`, 4