Package ‘fitbitr’
August 22, 2021

Type Package
Title Interface with the 'Fitbit' API
Version 0.2.0
Description Many 'Fitbit' users, and R-friendly 'Fitbit' users especially, have found themselves curious about their 'Fitbit' data. 'Fitbit' aggregates a large amount of personal data, much of which is interesting for personal research and to satisfy curiosity, and is even potentially useful in medical settings. The goal of 'fitbitr' is to make interfacing with the 'Fitbit' API as streamlined as possible, to make it simple for R users of all backgrounds and comfort levels to analyze their 'Fitbit' data and do whatever they want with it! Currently, 'fitbitr' includes methods for pulling data on activity, sleep, and heart rate, but this list is likely to grow in the future as the package gains more traction and more requests for new methods to be implemented come in. You can find details on the 'Fitbit' API at <https://dev.fitbit.com/build/reference/web-api/>.

License GPL (>= 3)
URL https://github.com/mrkaye97/fitbitr,
https://mrkaye97.github.io/fitbitr/

BugReports https://github.com/mrkaye97/fitbitr/issues
Imports dplyr, httr, janitor, jsonlite, lubridate, magrittr, purrr, rlang, tibble (>= 2.0.0), tidyr
Suggests covr, checkmate (>= 2.0.0), spelling, testthat (>= 3.0.0)

Config/testthat/edition 3
Encoding UTF-8
Language en-US
RoxygenNote 7.1.1
NeedsCompilation no
Author Matt Kaye [aut, cre]
Maintainer Matt Kaye <mrkaye97@gmail.com>
Repository CRAN
Date/Publication 2021-08-22 04:10:07 UTC
R topics documented:

activity_calories .............................................. 2
activity_summary ............................................. 3
calories ......................................................... 3
calories_bmr .................................................... 4
distance ......................................................... 5
elevation ......................................................... 5
floors ............................................................. 6
generate_token .................................................. 7
heart_rate_intraday ............................................. 8
heart_rate_zones ............................................... 9
lifetime_bests ................................................. 9
lifetime_totals ............................................... 10
loadCached_token ............................................. 10
minutes_fairly_active ........................................ 11
minutes_lightly_active ...................................... 11
minutes_sedentary ............................................. 12
minutes_very_active ......................................... 13
sleep_stage_granular ........................................ 13
sleep_stage_summary ......................................... 14
sleep_summary ................................................. 15
steps ............................................................. 15
tracker_bests ................................................. 16
tracker_totals ............................................... 16

Index 18

activity_calories Activity Calories Time Series

Description

Resource path /activities/activityCalories

Usage

activity_calories(start_date, end_date)

Arguments

start_date The start date of records to be returned in "yyyy-mm-dd" or date(time) format
end_date The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble with two columns: date and activity_calories
activity_summary

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
activity_calories(date)

## End(Not run)
```

---

activity_summary  

### Activity Summary

#### Description


#### Usage

```r
activity_summary(date)
```

#### Arguments

- **date**
  
  The date of records to be returned in "yyyy-mm-dd" or date(time) format

#### Value

A tibble with the date and a number of activity summary metrics for the day.

#### Examples

```r
## Not run:
date <- lubridate::today()
activity_summary(date)

## End(Not run)
```

---

calories  

### Calories Time Series

#### Description

Resource path /activities/calories

#### Usage

```r
calories(start_date, end_date)
```
Arguments

start_date  The start date of records to be returned in "yyyy-mm-dd" or date(time) format
end_date  The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble with two columns: date and calories

Examples

## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
calories(date)

## End(Not run)

---

**calories_bmr**  
*Calories BMR Time Series*

Description

Resource path /activities/caloriesBMR

Usage

```r
calories_bmr(start_date, end_date)
```

Arguments

start_date  The start date of records to be returned in "yyyy-mm-dd" or date(time) format
end_date  The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble with two columns: date and calories_bmr

Examples

## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
calories_bmr(date)

## End(Not run)
distance        *Distance Time Series*

**Description**

Resource path /activities/distance

**Usage**

distance(start_date, end_date)

**Arguments**

- **start_date**: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- **end_date**: The end date of records to be returned in "yyyy-mm-dd" or date(time) format

**Value**

A tibble with two columns: date and distance

**Examples**

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
distance(date)
## End(Not run)
```

---

elevation      *Elevation Time Series*

**Description**

Resource path /activities/elevation

**Usage**

elevation(start_date, end_date)

**Arguments**

- **start_date**: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- **end_date**: The end date of records to be returned in "yyyy-mm-dd" or date(time) format
floors

Value

A tibble with two columns: date and elevation

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
elevation(date)

## End(Not run)
```

floors                         Floors Time Series

Description

Resource path /activities/floors

Usage

```r
floors(start_date, end_date)
```

Arguments

- `start_date`: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`: The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble with two columns: date and floors

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
floors(date)

## End(Not run)
```
**generate_token**

*Generate a Fitbit API token*

**Description**

Simplify the setup process by persisting your Fitbit client_id and secret in the `.fitbitr-oauth` file.

**Usage**

```r
generate_token(
  client_id,  
  client_secret,  
  callback = "http://localhost:1410/",  
  scope = c("sleep", "activity", "heartrate", "location", "nutrition", "profile",  
               "settings", "social", "weight"),  
  cache = FALSE,  
  use_basic_auth = TRUE,  
  ...
)
```

**Arguments**

- `client_id`  
  your Fitbit client ID
- `client_secret`  
  your Fitbit client secret
- `callback`  
  your Fitbit redirect URL
- `scope`  
  the scopes to enable
- `cache`  
  Do you want to cache your token? See `oauth2.0_token` for details
- `use_basic_auth`  
  A boolean for whether or not to use basic auth in `oauth2.0_token`. Defaults to TRUE
- `...`  
  additional arguments to be passed to `oauth2.0_token`

**Value**

No return value. This function generates a token and saves it (hidden) in the global environment to be used for the remainder of the R session. You can cache this token with `cache = TRUE` or explicitly setting a filepath to cache to. See `oauth2.0_token` for details.

**Examples**

```r
## Not run:
generate_token(
  client_id = <YOUR-CLIENT-ID>  
  client_secret = <YOUR-CLIENT-SECRET>,  
  cache = TRUE  
)

## End(Not run)
```
heart_rate_intraday

Description

Returns heart rate data for the specified day

Usage

heart_rate_intraday(date, minutes = TRUE)

Arguments

date The start date of records to be returned in "yyyy-mm-dd" or date(time) format
minutes a boolean for whether data should be returned in minutes (TRUE) or seconds (FALSE)

Details


Value

A tibble of the time and your heart_rate at that time.

Examples

## Not run:
date <- lubridate::today()

## get minute by minute data
heart_rate_intraday(date, minutes = TRUE)

## get more granular data
## (not necessarily by second, but more granular than minutes)
heart_rate_intraday(date, minutes = FALSE)

## End(Not run)
heart_rate_zones  

Heart Rate Zones

Description


Usage

```r
heart_rate_zones(start_date, end_date = start_date)
```

Arguments

- `start_date`: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`: The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble of the date, the heart rate zone (`zone`), the minimum heart rate in that zone (`min_hr`), the maximum heart rate in that zone (`max_hr`), the minutes in that zone (`minutes_in_zone`), and the calories burned in that zone (`calories_out`)

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
heart_rate_zones(start_date, end_date)
## End(Not run)
```

lifetime_bests  

Lifetime Bests

Description

Retrieve lifetime best distance, floors, and steps

Usage

```r
lifetime_bests()
```

Value

A tibble the best distance, floors, and steps (by date) tracked on any of your trackers
## Examples

```r
## Not run:
lifetime_bests()

## End(Not run)
```

---

### lifetime_totals

#### Description

Retrieve lifetime total distance, floors, and steps

#### Usage

```r
lifetime_totals()
```

#### Value

A tibble of all-time totals across trackers (i.e. the total distance, floors, and steps tracked across all of your trackers)

### Examples

```r
## Not run:
lifetime_totals()

## End(Not run)
```

---

### load_cached_token

#### Description

Load a token from the cache

#### Usage

```r
load_cached_token(path = ".httr-oauth")
```

#### Arguments

- **path**
  
  the path to the file where the token is stored

#### Value

No return value. The token is stored in the global environment as a hidden variable named `.fitbitr_token`
minutes_fairly_active

Examples

```r
## Not run:
load_cached_token()

## End(Not run)
```

---

**minutes_fairly_active**  *Minutes Fairly Active Time Series*

**Description**

Resource path /activities/minutesFairlyActive

**Usage**

```r
minutes_fairly_active(start_date, end_date)
```

**Arguments**

- `start_date`: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`: The end date of records to be returned in "yyyy-mm-dd" or date(time) format

**Value**

A tibble with two columns: date and minutes_fairly_active

**Examples**

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
minutes_fairly_active(date)

## End(Not run)
```

---

**minutes_lightly_active**  *Minutes Lightly Active Time Series*

**Description**

Resource path /activities/minutesLightlyActive

**Usage**

```r
minutes_lightly_active(start_date, end_date)
```

**Examples**

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
minutes_lightly_active(date)

## End(Not run)
```
Arguments

- **start_date**: The start date of records to be returned in "yyyy-mm-dd" or date(time) format.
- **end_date**: The end date of records to be returned in "yyyy-mm-dd" or date(time) format.

Value

A tibble with two columns: date and minutes_lightly_active

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
minutes_lightly_active(date)
## End(Not run)
```

---

### minutes_sedentary

**Minutes Sedentary Time Series**

Description

Resource path /activities/minutesSedentary

Usage

```r
minutes_sedentary(start_date, end_date)
```

Arguments

- **start_date**: The start date of records to be returned in "yyyy-mm-dd" or date(time) format.
- **end_date**: The end date of records to be returned in "yyyy-mm-dd" or date(time) format.

Value

A tibble with two columns: date and minutes_sedentary

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
minutes_sedentary(date)
## End(Not run)
```
minutes_very_active  Minutes Very Active Time Series

Description

Resource path /activities/minutesVeryActive

Usage

minutes_very_active(start_date, end_date)

Arguments

start_date  The start date of records to be returned in "yyyymm-dd" or date(time) format
end_date  The end date of records to be returned in "yyyymm-dd" or date(time) format

Value

A tibble with two columns: date and minutes_very_active

Examples

## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
minutes_very_active(date)
## End(Not run)

sleep_stage_granular  Granular Sleep Stage Data

Description

Returns a tibble of nightly sleep stage data. Very granular. Returns blocks of time spent in each phase.

Usage

sleep_stage_granular(start_date, end_date = start_date)

Arguments

start_date  The start date of records to be returned in "yyyymm-dd" or date(time) format
end_date  The end date of records to be returned in "yyyymm-dd" or date(time) format
sleep_stage_summary

Value

A tibble of granular sleep stage data. This method is more granular than sleep_stage_summary, and returns blocks of time that you spent in each zone throughout the night.

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()

sleep_stage_granular(start_date, end_date)

## End(Not run)
```

---

sleep_stage_summary  
Nightly Sleep Stage Summary Data

Description

Returns a tibble of nightly sleep stage data. Minutes in each stage, count of times in each stage, and a thirty day average for the number of minutes in each stage.

Usage

```r
sleep_stage_summary(start_date, end_date = start_date)
```

Arguments

- `start_date`  
The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`  
The end date of records to be returned in "yyyy-mm-dd" or date(time) format

Value

A tibble of a variety of sleep stage summary data, by day

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()

sleep_stage_summary(start_date, end_date)

## End(Not run)
```
sleep_summary

**Nightly Sleep Summary**

**Description**

Returns a tibble of summary by night

**Usage**

```r
sleep_summary(start_date, end_date = start_date)
```

**Arguments**

- `start_date`: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`: The end date of records to be returned in "yyyy-mm-dd" or date(time) format

**Value**

A tibble of a variety of sleep summary data by day

**Examples**

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
sleep_summary(start_date, end_date)
## End(Not run)
```

---

steps

**Steps Time Series**

**Description**

Resource path /activities/steps

**Usage**

```r
steps(start_date, end_date)
```

**Arguments**

- `start_date`: The start date of records to be returned in "yyyy-mm-dd" or date(time) format
- `end_date`: The end date of records to be returned in "yyyy-mm-dd" or date(time) format
Value

A tibble with two columns: date and steps

Examples

```r
## Not run:
start_date <- lubridate::today() - lubridate::weeks(1)
end_date <- lubridate::today()
steps(date)

## End(Not run)
```

---

**tracker_bests**

**Tracker Bests**

Description

Retrieve tracker best distance, floors, and steps

Usage

`tracker_bests()`

Value

A tibble the best distance, floors, and steps (by date) tracked on your tracker

Examples

```r
## Not run:
tacker_bests()

## End(Not run)
```

---

**tracker_totals**

**Tracker Totals**

Description

Retrieve tracker total distance, floors, and steps

Usage

`tracker_totals()`
Value

A tibble of all-time tracker totals (i.e. the total distance, floors, and steps tracked by your tracker)

Examples

```r
## Not run:
tracker_totals()

## End(Not run)
```
Index

activity_calories, 2
activity_summary, 3

calories, 3
calories_bmr, 4
distance, 5
elevation, 5
floors, 6
generate_token, 7
heart_rate_intraday, 8
heart_rate_zones, 9
lifetim_bests, 9
lifetime_totals, 10
load_cached_token, 10
minutes_fairly_active, 11
minutes_lightly_active, 11
minutes_sedentary, 12
minutes_very_active, 13

oauth2.0_token, 7

sleep_stage_granular, 13
sleep_stage_summary, 14
sleep_summary, 15
steps, 15

tracker_best, 16
tracker_totals, 16