

# Package ‘fitbitr’

October 13, 2022

**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
load_cached_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
<b>Index</b>	<b>18</b>

---

activity_calories	<i>Activity Calories Time Series</i>
-------------------	--------------------------------------

---

**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

**Examples**

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

## End(Not run)
```

---

activity_summary	<i>Activity Summary</i>
------------------	-------------------------

---

**Description**

See <https://dev.fitbit.com/build/reference/web-api/activity/> for more details.

**Usage**

```
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**

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

## End(Not run)
```

---

calories	<i>Calories Time Series</i>
----------	-----------------------------

---

**Description**

Resource path /activities/calories

**Usage**

```
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	<i>Calories BMR Time Series</i>
--------------	---------------------------------

---

**Description**

Resource path /activities/caloriesBMR

**Usage**

```
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	<i>Distance Time Series</i>
----------	-----------------------------

---

**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**

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

---

elevation	<i>Elevation Time Series</i>
-----------	------------------------------

---

**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

**Value**

A tibble with two columns: date and elevation

**Examples**

```
## 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**

```
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**

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

---

generate_token	<i>Generate a Fitbit API token</i>
----------------	------------------------------------

---

### Description

Simplify the setup process by persisting your Fitbit client\_id and secret in the .fitbitr-oauth file.

### Usage

```
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 <a href="#">oauth2.0_token</a> for details
use_basic_auth	A boolean for whether or not to use basic auth in <a href="#">oauth2.0_token</a> . Defaults to TRUE
...	additional arguments to be passed to <a href="#">oauth2.0_token</a>

### 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

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

---

heart\_rate\_intraday *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

See <https://dev.fitbit.com/reference/web-api/heart-rate/#get-heart-rate-time-series> for more 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	<i>Heart Rate Zones</i>
------------------	-------------------------

---

**Description**

See <https://dev.fitbit.com/build/reference/web-api/activity/> for more details.

**Usage**

```
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**

```
## 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	<i>Lifetime Bests</i>
----------------	-----------------------

---

**Description**

Retrieve lifetime best distance, floors, and steps

**Usage**

```
lifetime_bests()
```

**Value**

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

**Examples**

```
## Not run:
lifetime_bests()

## End(Not run)
```

---

lifetime_totals	<i>Lifetime Totals</i>
-----------------	------------------------

---

**Description**

Retrieve lifetime total distance, floors, and steps

**Usage**

```
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**

```
## Not run:
lifetime_totals()

## End(Not run)
```

---

load_cached_token	<i>Load a token from the cache</i>
-------------------	------------------------------------

---

**Description**

Load a token from the cache

**Usage**

```
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`

**Examples**

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

---

minutes\_fairly\_active *Minutes Fairly Active Time Series*

---

**Description**

Resource path /activities/minutesFairlyActive

**Usage**

```
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**

```
## 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**

```
minutes_lightly_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\_lightly\_active

**Examples**

```
## 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**

```
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**

```
## 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 "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\_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 "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 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**

```
## 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**

```
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**

```
## 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	<i>Nightly Sleep Summary</i>
---------------	------------------------------

---

**Description**

Returns a tibble of summary by night

**Usage**

```
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**

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

---

steps	<i>Steps Time Series</i>
-------	--------------------------

---

**Description**

Resource path /activities/steps

**Usage**

```
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**

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

---

tracker_bests	<i>Tracker Bests</i>
---------------	----------------------

---

**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**

```
## Not run:  
tracker_bests()  
  
## End(Not run)
```

---

tracker_totals	<i>Tracker Totals</i>
----------------	-----------------------

---

**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**

```
## 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

[lifetime\\_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, 14  
[sleep\\_summary](#), 15  
[steps](#), 15

[tracker\\_bests](#), 16  
[tracker\\_totals](#), 16