

Package ‘rwalkr’

December 13, 2018

Type Package

Title API to Melbourne Pedestrian Data

Version 0.4.0

Description Provides API to Melbourne pedestrian data in tidy data form.

Depends R (>= 3.1.3)

Imports tidyr, dplyr, httr

Suggests shiny (>= 1.0.4), plotly

URL <http://pkg.earo.me/rwalkr>

BugReports <https://github.com/earowang/rwalkr/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Earo Wang [aut, cre] (<<https://orcid.org/0000-0001-6448-5260>>)

Maintainer Earo Wang <earo.wang@gmail.com>

Repository CRAN

Date/Publication 2018-12-13 05:50:03 UTC

R topics documented:

lookup_sensor	2
melb_shine	2
melb_walk	3
melb_walk_fast	4
pull_sensor	5

Index	7
--------------	----------

lookup_sensor	<i>Look up sensor names between melb_walk_fast() and melb_walk()</i>
---------------	--

Description

One-to-one corresponding sensor names between melb_walk_fast() and melb_walk()

Usage

```
lookup_sensor()
```

Details

Two APIs (Socrata and compedapi) code some sensors using different names. This functions returns a data frame that allows to compare sensor names obtained from these two APIs.

Value

A data frame including three columns:

- run: Sensor names obtained from the melb_walk_fast() function using Socrata
- walk: Sensor names obtained from the melb_walk() function using compedapi
- match: whether sensor names are identical or not

Examples

```
lookup_sensor()
```

melb_shine	<i>A simple shiny app for pedestrian data</i>
------------	---

Description

Provides a GUI to download data of selected sensors over a specified period as a CSV file, accompanied with basic visualisation.

Usage

```
melb_shine()
```

Details

It offers some basic plots to give a glimpse of the data over a short time period. In order to be reproducible, scripting using walk_melb or run_melb is recommended.

Value

A shiny app.

See Also

[melb_walk](#), [melb_walk_fast](#)

melb_walk

API using compedapi to Melbourne pedestrian data

Description

Provides API using compedapi to Melbourne pedestrian data in a tidy data form.

Usage

```
melb_walk(from = to - 6L, to = Sys.Date() - 1L, tz = "",  
          na.rm = FALSE, session = NULL)
```

Arguments

from	Starting date.
to	Ending date.
tz	Deprecated. For this dataset, it should only be "Australia/Melbourne".
na.rm	Logical. FALSE is the default suggesting to include NA in the dataset. TRUE removes the NAs.
session	NULL or "shiny". For internal use only.

Details

It provides API using compedapi, where counts are uploaded on a daily basis. The up-to-date data would be till the previous day. The data is sourced from [Melbourne Open Data Portal](#). Please refer to Melbourne Open Data Portal for more details about the dataset and its policy.

Value

A tibble including these variables as follows:

- Sensor: Sensor name (43 sensors up to date)
- Date_Time: Date time when the pedestrian counts are recorded
- Date: Date associated with Date_Time
- Time: Time of day
- Count: Hourly counts

See Also[melb_walk_fast](#)**Examples**

```
## Not run:
# Retrieve last week data
melb_walk()

# Retrieve data of a specified period
start_date <- as.Date("2017-07-01")
end_date <- start_date + 6L
melb_walk(from = start_date, to = end_date)

## End(Not run)
```

`melb_walk_fast`*API using Socrata to Melbourne pedestrian data*

Description

Provides API using Socrata to Melbourne pedestrian data in a tidy data form.

Usage

```
melb_walk_fast(year = NULL, sensor = NULL, tz = "", na.rm = FALSE,
  app_token = NULL)
```

Arguments

<code>year</code>	An integer or a vector of integers. By default, it's the current year.
<code>sensor</code>	Sensor names. By default, it pulls all the sensors. Use lookup_sensor to see the available sensors.
<code>tz</code>	Deprecated. For this dataset, it should only be "Australia/Melbourne".
<code>na.rm</code>	Logical. FALSE is the default suggesting to include NA in the dataset. TRUE removes the NAs.
<code>app_token</code>	Characters giving the application token. A limited number of requests can be made without an app token (NULL), but they are subject to much lower throttling limits than request that do include one. Sign up for an app token here .

Details

It provides API using [Socrata](#), where counts are uploaded on a monthly basis. The up-to-date data would be till the previous month. The data is sourced from [Melbourne Open Data Portal](#). Please refer to Melbourne Open Data Portal for more details about the dataset and its policy.

Value

A tibble including these variables as follows:

- Sensor: Sensor name (46 sensors up to date)
- Date_Time: Date time when the pedestrian counts are recorded
- Date: Date associated with Date_Time
- Time: Time of day
- Count: Hourly counts

See Also

[melb_walk](#)

Examples

```
## Not run:  
# Retrieve the year 2017  
melb_walk_fast(year = 2017)  
  
# Retrieve the year 2017 for Southern Cross Station  
melb_walk_fast(year = 2017, sensor = "Southern Cross Station")  
  
## End(Not run)
```

pull_sensor

API using Socrata to Melbourne pedestrian sensor locations

Description

Provides API using Socrata to Melbourne pedestrian sensor locations.

Usage

```
pull_sensor(app_token = NULL)
```

Arguments

app_token Characters giving the application token. A limited number of requests can be made without an app token (NULL), but they are subject to much lower throttling limits than request that do include one. Sign up for an app token [here](#).

Details

It provides API using [Socrata](#).

Value

A data frame including these variables as follows:

- Sensor: Sensor name (43 sensors up to date)
- Sensor_ID: Sensor identifier
- Longitude: Longitude
- Latitude: Latitude
- Location_Type: Location type
- Year_Installed: Year installed

See Also

[melb_walk_fast](#)

Examples

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

Index

lookup_sensor, [2](#), [4](#)

melb_shine, [2](#)

melb_walk, [3](#), [3](#), [5](#)

melb_walk_fast, [3](#), [4](#), [4](#), [6](#)

pull_sensor, [5](#)