

Package ‘jpstat’

February 11, 2023

Type Package

Title Tools for Easy Use of 'e-Stat', 'RESAS' API, Etc

Version 0.3.2

Description Provides tools to use API such as 'e-Stat' (<<https://www.e-stat.go.jp/>>), the portal site for Japanese government statistics, and 'RESAS' (Regional Economy and Society Analyzing System, <<https://resas.go.jp>>).

License MIT + file LICENSE

URL <https://github.com/UchidaMizuki/jpstat>,
<https://uchidamizuki.github.io/jpstat/>

BugReports <https://github.com/UchidaMizuki/jpstat/issues>

Encoding UTF-8

Suggests keyring (>= 1.2.0), knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

Imports dplyr (>= 0.8.0), httr, pillar, purrr (>= 1.0.0), rlang (>= 0.3.0), stringr (>= 1.3.0), tibble (>= 1.3.1), tidyr (>= 1.0.0), vctrs, navigatr (>= 0.2.0), lifecycle, sticky

Depends R (>= 4.1.0)

NeedsCompilation no

Author Mizuki Uchida [aut, cre]

Maintainer Mizuki Uchida <uchidamizuki@vivaldi.net>

Repository CRAN

Date/Publication 2023-02-11 10:30:02 UTC

R topics documented:

estat	2
estat_table_info	3
jpstat	3
resas	4
webland	4

estat	<i>Access 'e-Stat' data</i>
-------	-----------------------------

Description

The `estat()` gets the meta-information of a statistical table by using `getMetaInfo` of the 'e-Stat' API, and returns an `estat` object that allows editing of meta-information by `dplyr::filter()` and `dplyr::select()`.

Usage

```
estat(  
  appId,  
  statsDataId,  
  lang = c("J", "E"),  
  query = list(),  
  path = "rest/3.0/app/json/"  
)
```

Arguments

<code>appId</code>	An 'appId' of 'e-Stat' API.
<code>statsDataId</code>	A statistical data ID on 'e-Stat'.
<code>lang</code>	A language, Japanese ("J") or English ("E").
<code>query</code>	A list of additional queries.
<code>path</code>	An e-Stat API path.

Value

A `estat` object.

See Also

<https://www.e-stat.go.jp>
<https://www.e-stat.go.jp/en>

Examples

```
## Not run:  
estat("Your appId", "https://www.e-stat.go.jp/dbview?sid=0003433219")  
  
## End(Not run)
```

estat_table_info	<i>Get table information for 'e-Stat' data</i>
------------------	--

Description

Get table information for 'e-Stat' data

Usage

```
estat_table_info(x)
```

Arguments

x A estat object.

Value

A tbl_df of the table information.

jpstat	<i>Tools for easy use of 'e-Stat' API</i>
--------	---

Description

Provides tools to use API such as 'e-Stat' (<https://www.e-stat.go.jp/>), the portal site for Japanese government statistics, and 'RESAS' (Regional Economy and Society Analyzing System, <https://resas.go.jp>).

Author(s)

Maintainer: Mizuki Uchida <uchidamizuki@vivaldi.net>

See Also

Useful links:

- <https://github.com/UchidaMizuki/jpstat>
- <https://uchidamizuki.github.io/jpstat/>
- Report bugs at <https://github.com/UchidaMizuki/jpstat/issues>

resas	<i>Access 'RESAS' data</i>
-------	----------------------------

Description

[Experimental]

Usage

```

resas(
  X_API_KEY,
  path,
  query = list(),
  to_snakecase = TRUE,
  names_sep = "/",
  rectangle = TRUE
)

```

Arguments

X_API_KEY	An 'X-API-KEY' of 'RESAS' API.
path	A 'RESAS' API path.
query	Additional queries.
to_snakecase	Whether the parameters and responses should be named as snake cases or not?
names_sep	A character that separates the names of the responses.
rectangle	Whether to rectangle the data or not?

Value

A resas object.

See Also

<https://opendata.resas-portal.go.jp/>

webland	<i>Information on real estate transaction prices API</i>
---------	--

Description

[Experimental]

Usage

```
webland_trade(lang = c("ja", "en"))
```

```
webland_city(lang = c("ja", "en"))
```

Arguments

lang Language.

Details

Collect data from the information on real estate transaction prices API provided by Japan's Ministry of Land, Infrastructure, Transport and Tourism.

Value

webland_trade() a webland_trade object. By creating a query with itemise() and applying collect(), The real estate transaction prices are collected.

webland_city() a webland_city object. Obtains a list of target municipalities in the same way as webland_trade().

Examples

```
## Not run:
# Collect trade data
webland_trade() |>
  itemise(from = "20151",
          to = "20152",
          city_code = "13102") |>
  collect()

# Collect target municipalities
webland_city() |>
  itemise(pref_code = "13") |>
  collect()

## End(Not run)
```

Index

estat, [2](#)

estat_table_info, [3](#)

jpstat, [3](#)

jpstat-package (jpstat), [3](#)

resas, [4](#)

webland, [4](#)

webland_city (webland), [4](#)

webland_trade (webland), [4](#)