Package 'danstat'

October 13, 2022

Type Package

Title R Client for the Statistics Denmark Databank API

Version 0.2.0

Author Valeri Voev

Maintainer Valeri Voev <v_voev@yahoo.com>

Description The purpose of the package is to enable an R function interface into the Statistics Denmark Databank API mainly for research purposes.
The Statistics Denmark Databank API has four endpoints, see here for more information and testing the API in their console: <https://www.dst.dk/en/Statistik/brug-statistikken/muligheder-i-statistikbanken/api>.
This package mimics the structure of the API and provides four main functions to match the functionality of the API endpoints.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports httr, jsonlite, readr

Suggests testthat (>= 2.1.0), knitr, rmarkdown, purrr, dplyr, ggplot2, kableExtra

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-01-31 08:10:02 UTC

R topics documented:

get_data *Get data for a particular table and variable selection*

Description

Get data for a particular table and variable selection

Usage

```
get_data(table_id, variables, language = c("en", "da"))
```

Arguments

table_id	Table identifier, e.g. "folk1a"
variables	A list with variable code-values pairs. Each code-values pair should be a named list with names "code" and "values". If all values for a variable are desired, define values = NA for that variable code.
language	Language for the return object. Default = "en"

Value

A data frame

Examples

```
# Get data from table "folk1c" for selected values of variables "ieland" and "køn"
# and all time periods available.

# The "ieland" variable is filtered for Denmark (id = 5000) and Bulgaria (id = 5128)
# and the "køn" variable id filtered for Men (id = 1) and Women (id = 2).
# The "tid" variable is unfiltered, i.e. selects all available time periods
# See get_table_metadata(table_id = "folk1c", variables_only = TRUE) for variable codes and values.

variables <- list(list(code = "ieland", values = c(5100, 5128)),
                  list(code = "køn", values = c(1,2)),
                  list(code = "tid", values = NA))

data <- get_data("folk1c", variables)
```

`get_subjects` *Get a list of subjects covered in the data bank*

Description

Get a list of subjects covered in the data bank

Usage

```
get_subjects(
  subjects = NULL,
  recursive = FALSE,
  include_tables = FALSE,
  language = c("en", "da")
)
```

Arguments

`subjects`	Provide specific subject id's to get subtopics. E.g. `subjects = c("02", "2419")`
`recursive`	Whether subtopics/tables will be retrieved all the way down the hierarchy. Otherwise, only the closest level under the provided subjects will be retrieved. Default = `FALSE`
`include_tables`	Whether the result should contain tables. Otherwise, only subjects are returned. Default = `FALSE`
`language`	Language for the return object. Default = `"en"`

Value

A data frame

Examples

```
# Get all subjects
all_subjects <- get_subjects()

# Or get (sub)subjects for specific subjects
some_subjects <- get_subjects(subjects = c("2", "3"))

# Get all subject hierarchy for a given subject
subject_with_hierarchy <- get_subjects(subjects = "2", recursive = TRUE)
```

get_tables *Get a list of stables in the data bank*

Description

Get a list of stables in the data bank

Usage

```
get_tables(
  subjects = NULL,
  pastdays = NA_integer_,
  include_inactive = FALSE,
  language = c("en", "da")
)
```

Arguments

subjects	Provide specific subject id's to get subtopics. E.g. subjects = c("02", "2419"). Can be retrieved with get_subjects()
pastdays	Return only tables which have been updated within this number of days
include_inactive	Whether to return tables that are no longer updated
language	Language for the return object. Default = "en"

Value

A data frame

Examples

```
# Get all tables
all_tables <- get_tables()

# Or get tables for specific subjects
some_tables <- get_tables(subjects = c("2", "3413"))

# Get all tables updated within the past 3 days
tables_past3days <- get_tables(pastdays = 3)
```

`get_table_metadata` *Title*

Description

Title

Usage

```
get_table_metadata(table_id, variables_only = FALSE, language = c("en", "da"))
```

Arguments

`table_id`	Table identifier, e.g. "folk1a"
`variables_only`	If `TRUE` returns only information about the variables in the table
`language`	Language for the return object. Default = `"en"`

Value

A list with information about the table, like documentation url, variable description, etc. If `variables_only = TRUE`, returns a data frame with variable information.

Examples

```
# Get table metadata for a given table
table_meta <- get_table_metadata(table_id = "folk1c") # a list

# Get only information about the variables in the table
table_meta_vars <- get_table_metadata(table_id = "folk1c", variables_only = TRUE) # a data frame
```

Index