

Package ‘cbsodataR’

February 21, 2019

Type Package

Title Statistics Netherlands (CBS) Open Data API Client

Version 0.3.2

Description The data and meta data from Statistics Netherlands (www.cbs.nl) can be browsed and downloaded. The client uses the open data API of Statistics Netherlands.

License GPL-2

URL <https://github.com/edwindj/cbsodataR>

BugReports <https://github.com/edwindj/cbsodataR/issues>

Imports whisker, jsonlite, utils

Suggests knitr, rmarkdown, dplyr, shiny

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Edwin de Jonge [aut, cre]

Maintainer Edwin de Jonge <edwindjonge@gmail.com>

Repository CRAN

Date/Publication 2019-02-21 15:40:03 UTC

R topics documented:

cache_clear	2
cbs_add_date_column	2
cbs_add_label_columns	3
cbs_default_selection	4
cbs_download_data	4
cbs_download_meta	5
cbs_download_table	5
cbs_extract_table_id	6
cbs_get_data	7

cbs_get_data_from_link	8
cbs_get_meta	9
cbs_get_meta_from_dir	9
cbs_get_tables_themes	10
cbs_get_themes	10
cbs_get_toc	11
download_data-deprecated	12
download_meta-deprecated	13
download_table	13
get_data	14
get_meta	15
get_meta_from_dir	16
get_tables_themes	16
get_table_list	17
get_themes	18
resolve_deeplink	19

Index **20**

cache_clear	<i>clears the cache</i>
-------------	-------------------------

Description

clears the cache

Usage

cache_clear()

cbs_add_date_column	<i>Convert the time variable into either a date or numeric.</i>
---------------------	---

Description

Time periods in data of CBS are coded: yyyyXXww (e.g. 2018JJ00, 2018MM10, 2018KW02), which contains year (yyyy), type (XX) and index (ww). `cbs_add_date_column` converts these codes into a [Date](#) or numeric. In addition it adds a frequency column denoting the type of the column.

Usage

`cbs_add_date_column(x, date_type = c("Date", "numeric"), ...)`

Arguments

x	data.frame retrieved using cbs_get_data
date_type	Type of date column: "Date", "numeric. Numeric creates a fractional number which signs the "middle" of the period. e.g. 2018JJ00 -> 2018.5 and 2018KW01 -> 2018.167. This is for the following reasons: otherwise 2018.0 could mean 2018, 2018 Q1 or 2018 Jan, and furthermore 2018.75 is a bit strange for 2018 Q4. If all codes in the dataset have frequency "Y" the numeric output will be integer.
...	future use.

Value

original dataset with two added columns: <period>_date and <period>_freq. This last column is a factor with levels: Y, Q and M

`cbs_add_label_columns` *For each column with codes add label column to data set*

Description

Adds cbs labels to the dataset that was retrieved using [cbs_get_data](#). Code columns will be translated into label columns for each of the column that was supplied. By default all code columns will be accompanied with a label column.

Usage

```
cbs_add_label_columns(x, columns = colnames(x), ...)
```

Arguments

x	data.frame retrieved using cbs_get_data .
columns	character with the names of the columns for which labels will be added
...	not used.

Examples

```
## Not run:

# get data for main (000000) Consumer Price Index (7196ENG) for March 2000,
x <- cbs_get_data(id="7196ENG", Periods="2000MM03", CPI="000000")
cbs_add_label_columns(x)

## End(Not run)
```

`cbs_default_selection` *extract the default selection from a cbsodata meta object*

Description

extract the default selection from a cbsodata meta object

Usage

```
cbs_default_selection(x, ...)
```

Arguments

<code>x</code>	meta object
<code>...</code>	for future use

`cbs_download_data` *Gets all data from a cbs table.*

Description

Gets all data via bulk download. `download_data` dumps the data in (international) csv format.

Usage

```
cbs_download_data(id, path = file.path(id, "data.csv"), ...,
  select = NULL, typed = TRUE, verbose = FALSE,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

<code>id</code>	of cbs open data table
<code>path</code>	of data file, defaults to "<id>/data.csv"
<code>...</code>	optional filter statements to select rows of the data,
<code>select</code>	optional names of columns to be returned.
<code>typed</code>	Should the data automatically be converted into integer and numeric?
<code>verbose</code>	show the underlying downloading of the data
<code>base_url</code>	optionally specify a different server. Useful for third party data services implementing the same protocol.

cbs_download_meta *Dumps the meta data into a directory*

Description

Dumps the meta data into a directory

Usage

```
cbs_download_meta(id, dir = id, ..., verbose = FALSE, cache = FALSE,  
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	Id of CBS open data table (see cbs_get_toc)
dir	Directory in which data should be stored. By default it creates a sub directory with the name of the id
...	not used
verbose	Print extra messages what is happening.
cache	Should meta data be cached?
base_url	optionally allow to specify a different server. Useful for third party data services implementing the same protocol.

Value

meta data object

cbs_download_table *Download a table from statistics Netherlands*

Description

Download a table from statistics Netherlands

Usage

```
cbs_download_table(id, ..., dir = id, cache = FALSE, verbose = TRUE,  
  typed = FALSE, base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	Identifier of CBS table (can be retrieved from cbs_get_toc)
...	Parameters passed on to cbs_download_data
dir	Directory where table should be downloaded
cache	If metadata is cached use that, otherwise download meta data
verbose	Print extra messages what is happening.
typed	Should the data automatically be converted into integer and numeric?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

cbs_download_table retrieves all raw meta data and data and stores these as csv files in the directory specified by dir. It is possible to add a filter. A filter is specified with <column_name> = <values> in which <values> is a character vector. Rows with values that are not part of the character vector are not returned.

Examples

```
## Not run:

# download meta data and data from inflation/Consumer Price Indices
download_table(id="7196ENG")

## End(Not run)
```

cbs_extract_table_id *extract the id of a cbs table from the statline url*

Description

extract the id of a cbs table from the statline url

Usage

```
cbs_extract_table_id(url, ...)
```

Arguments

url	character with hyperlink to StatLine table
...	future use.

Value

character with id, will be NA if not found.

`cbs_get_data`*Get data from Statistics Netherlands (CBS)*

Description

Retrieves data from a table of Statistics Netherlands. A list of available tables can be retrieved with `cbs_get_toc`. Use the Identifier column of `cbs_get_toc` as `id` in `cbs_get_data` and `cbs_get_meta`. Optionally the data can be filtered on category values. The filter is specified with `<column_name> = <values>` in which `<values>` is a character vector. Rows with values that are not part of the character vector are not returned. Note that the values have to be raw (un-recorded) values.

Usage

```
cbs_get_data(id, ..., select = NULL, typed = TRUE,
             add_column_labels = TRUE, dir = tempdir(), verbose = FALSE,
             base_url = getOption("cbsodataR.base_url", BASE_URL),
             include_ID = FALSE)
```

Arguments

<code>id</code>	Identifier of table, can be found in <code>cbs_get_toc</code>
<code>...</code>	optional filter statements
<code>select</code>	character optional, columns to select
<code>typed</code>	Should the data automatically be converted into integer and numeric?
<code>add_column_labels</code>	Should column titles be added as a label (TRUE) which are visible in View
<code>dir</code>	Directory where the table should be downloaded. Defaults to temporary directory
<code>verbose</code>	Print extra messages what is happening.
<code>base_url</code>	optionally specify a different server. Useful for third party data services implementing the same protocol.
<code>include_ID</code>	Should the data include the ID column for the rows?

Details

By default the columns will be converted to their type (`typed=TRUE`). CBS uses multiple types of missing (unknown, suppressed, not measured, missing): users wanting all these nuances can use `typed=FALSE` which results in character columns.

Value

`data.frame` with the requested data. Note that a csv copy of the data is stored in `dir`.

Note

All data are downloaded using [cbs_download_table](#)

See Also

[cbs_get_meta](#), [cbs_download_data](#)

Examples

```
## Not run:

# get data for main (000000) Consumer Price Index (7196ENG) for March 2000,
  cbs_get_data(id="7196ENG", Periods="2000MM03", CPI="000000")

## End(Not run)
```

cbs_get_data_from_link

Retrieve data from a

Description

Retrieve data from a

Usage

```
cbs_get_data_from_link(link, message = TRUE, ...,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

link	url/hyperlink to opendata table made with the StatLine App
message	logical Should the query be printed (default TRUE)
...	passed on to cbs_get_data
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

Same as [cbs_get_data](#)

cbs_get_meta	<i>Get metadata of a cbs table</i>
--------------	------------------------------------

Description

Retrieve the meta data of a CBS open data table. Caching (cache=TRUE) improves the performance considerably. The meta data of a CBS table is

Usage

```
cbs_get_meta(id, verbose = FALSE, cache = TRUE,
             base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	internal id of CBS table, can be retrieved with cbs_get_toc
verbose	Print extra messages what is happening.
cache	should the result be cached?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

cbs_table object containing several data.frames with meta data.

cbs_get_meta_from_dir	<i>Load meta data from a downloaded table</i>
-----------------------	---

Description

Load meta data from a downloaded table

Usage

```
cbs_get_meta_from_dir(dir)
```

Arguments

dir	Directory where data was downloaded
-----	-------------------------------------

Value

cbs_table object with meta data

`cbs_get_tables_themes` *Get a the list of tables connected to themes*

Description

Get a the list of tables connected to themes

Usage

```
cbs_get_tables_themes(..., select = NULL, verbose = FALSE,
  cache = TRUE, base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

<code>...</code>	Use this to add a filter to the query e.g. <code>get_tables_themes(ID=10)</code> .
<code>select</code>	character vector with names of wanted properties. default is all
<code>verbose</code>	Print extra messages what is happening.
<code>cache</code>	Should the result be cached?
<code>base_url</code>	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

A `data.frame` with various properties of SN/CBS themes.

`cbs_get_themes` *Get list of all cbs thematic entries.*

Description

Returns a list of all cbs themes.

Usage

```
cbs_get_themes(..., select = NULL, verbose = TRUE, cache = FALSE,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

<code>...</code>	Use this to add a filter to the query e.g. <code>get_themes(ID=10)</code> .
<code>select</code>	character vector with names of wanted properties. default is all
<code>verbose</code>	Print extra messages what is happening.
<code>cache</code>	Should the result be cached?
<code>base_url</code>	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

A data.frame with various properties of SN/CBS themes.

The filter is specified with `<column_name> = <values>` in which `<values>` is a character vector.

Rows with values that are not part of the character vector are not returned.

Examples

```
## Not run:
# get list of all themes
cbs+get_themes()

# get list of all dutch themes from the Catalog "CBS"
cbs_get_themes(Language="nl", Catalog="CBS")

## End(Not run)
```

cbs_get_toc

Retrieve a data.frame with requested cbs tables

Description

`cbs_get_toc` by default a list of all tables and all columns will be retrieved. You can restrict the query by supplying multiple filter statements or by specifying the columns that should be returned.

Usage

```
cbs_get_toc(..., convert_dates = TRUE, select = NULL,
  verbose = FALSE, cache = TRUE,
  base_url = getOption("cbsodataR.base_url", BASE_URL),
  include_ID = FALSE)
```

Arguments

<code>...</code>	filter statement to select rows, e.g. <code>Language="nl"</code>
<code>convert_dates</code>	convert the columns with date-time information into <code>DateTime</code> (default <code>TRUE</code>)
<code>select</code>	character columns to be returned, by default all columns will be returned.
<code>verbose</code>	logical prints the calls to the webservice
<code>cache</code>	logical should the result be cached?
<code>base_url</code>	optionally specify a different server. Useful for third party data services implementing the same protocol.
<code>include_ID</code>	logical column needed by OData but with no current use.

Value

data.frame with identifiers, titles and descriptions of tables

Note

cbs_get_toc will cache results, so subsequent calls will be much faster.

Examples

```
## Not run:

# get list of english tables
tables_en <- cbs_get_toc(Language="en")

# get list of dutch tables
tables_nl <- cbs_get_toc(Language="nl")
View(tables_nl)

## End(Not run)
```

download_data-deprecated

Gets all data from a cbs table.

Description

Gets all data via bulk download. download_data dumps the data in (international) csv format.

Usage

```
download_data(id, path = file.path(id, "data.csv"), ..., select = NULL,
  typed = FALSE, verbose = TRUE,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	of cbs open data table
path	of data file, defaults to "<id>/data.csv"
...	optional filter statements to select rows of the data,
select	optional names of columns to be returned.
typed	Should the data automatically be converted into integer and numeric?
verbose	show the underlying downloading of the data
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

 download_meta-deprecated

Dumps the meta data into a directory

Description

Dumps the meta data into a directory

Usage

```
download_meta(id, dir = id, ..., verbose = FALSE, cache = FALSE,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	Id of CBS open data table
dir	Directory in which data should be stored. By default it creates a sub directory with the name of the id
...	not used
verbose	Print extra messages what is happening.
cache	Should meta data be cached?
base_url	optionally allow to specify a different server. Useful for third party data services implementing the same protocol.

Value

meta data object

 download_table

Download a table from statistics Netherlands

Description

Download a table from statistics Netherlands

Usage

```
download_table(id, ..., dir = id, cache = FALSE, verbose = TRUE,
  typed = FALSE, base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	Identifier of CBS table (can be retrieved from get_table_list)
...	Parameters passed on to download_data
dir	Directory where table should be downloaded
cache	If metadata is cached use that, otherwise download meta data
verbose	Print extra messages what is happening.
typed	Should the data automatically be converted into integer and numeric?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

download_table retrieves all raw meta data and data and stores these as csv files in the directory specified by dir. It is possible to add a filter. A filter is specified with <column_name> = <values> in which <values> is a character vector. Rows with values that are not part of the character vector are not returned.

Examples

```
## Not run:

# download meta data and data from inflation/Consumer Price Indices
download_table(id="7196ENG")

## End(Not run)
```

get_data

Get data from Statistics Netherlands (CBS)

Description

Retrieves data from a table of Statistics Netherlands. A list of tables can be retrieved with [get_table_list](#). Optionally the data can be filtered on category values. The filter is specified with <column_name> = <values> in which <values> is a character vector. Rows with values that are not part of the character vector are not returned. Note that the values have to be raw (un-recoded) values.

Usage

```
get_data(id, ..., recode = TRUE, use_column_title = recode,
         dir = tempdir(), base_url = getOption("cbsodataR.base_url",
         BASE_URL))
```

Arguments

id	Identifier of table, can be found in get_table_list
...	optional filter statemenets
recode	Should the categories of the table be recoded with their title (TRUE) or with their key (FALSE)?

use_column_title	Should column names be coded with title (TRUE) or key (FALSE)
dir	Directory where the table should be downloaded. Defaults to temporary directory
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

data.frame with the requested data. Note that a csv copy of the data is stored in dir.

Note

All data are downloaded using [download_table](#)

Examples

```
## Not run:

# get data for main (000000) Consumer Price Index (7196ENG) for March 2000,
get_data(id="7196ENG", Periods="2000MM03", CPI="000000")

## End(Not run)
```

get_meta	<i>Get metadata of a cbs table</i>
----------	------------------------------------

Description

Retrieve the meta data of a CBS open data table. Caching (cache=TRUE) may improve the performance considerably. The meta data of a CBS table is

Usage

```
get_meta(id, verbose = TRUE, cache = FALSE,
         base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

id	internal id of CBS table, can be retrieved with get_table_list
verbose	Print extra messages what is happening.
cache	should the result be cached?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

cbs_table object containing several data.frames with meta data.

get_meta_from_dir *Load meta data from a downloaded table*

Description

Load meta data from a downloaded table

Usage

```
get_meta_from_dir(dir)
```

Arguments

dir Directory where data was downloaded

Value

cbs_table object with meta data

get_tables_themes *Get a the list of tables connected to themes*

Description

Get a the list of tables connected to themes

Usage

```
get_tables_themes(..., select = NULL,
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

... Use this to add a filter to the query e.g. get_tables_themes(ID=10).

select character vector with names of wanted properties. default is all

base_url optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

A data.frame with various properties of SN/CBS themes.

get_table_list	<i>Retrieve a data.frame with requested cbs tables</i>
----------------	--

Description

get_table_list by default a list of all tables and all columns will be retrieved. You can restrict the query by supplying multiple filter statements or by specifying the columns that should be returned.

Usage

```
get_table_list(..., select = NULL,  
               base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

...	filter statement to select rows, e.g. Language="nl"
select	character columns to be returned, by default all columns will be returned.
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

data.frame with identifiers, titles and descriptions of tables

Note

get_table_list will cache results, so subsequent calls will be much faster.

Examples

```
## Not run:  
  
# get list of english tables  
tables_en <- get_table_list(Language="en")  
  
# get list of dutch tables  
tables_nl <- get_table_list(Language="nl")  
View(tables_nl)  
  
## End(Not run)
```

get_themes	<i>Get list of all cbs thematic entries.</i>
------------	--

Description

Returns a list of all cbs themes.

Usage

```
get_themes(..., select = NULL, verbose = TRUE, cache = FALSE,  
            base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

...	Use this to add a filter to the query e.g. <code>get_themes(ID=10)</code> .
select	character vector with names of wanted properties. default is all
verbose	Print extra messages what is happening.
cache	Should the result be cached?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

A data.frame with various properties of SN/CBS themes.

The filter is specified with `<column_name> = <values>` in which `<values>` is a character vector. Rows with values that are not part of the character vector are not returned.

Examples

```
## Not run:  
# get list of all themes  
get_themes()  
  
# get list of all dutch themes from the Catalog "CBS"  
get_themes(Language="nl", Catalog="CBS")  
  
## End(Not run)
```

resolve_deeplink	<i>resolve a deeplink created in the opendata portal</i>
------------------	--

Description

resolve a deeplink created in the opendata portal

Usage

```
resolve_deeplink(deeplink, ...,  
  base_url = getOption("cbsodataR.base_url", BASE_URL))
```

Arguments

deeplink	url to the deeplink in the opendataportal
...	used in the query
base_url	optionally specify a different server. Useful for third party data services implementing the same protocol.

Value

information object with table id, select, filter and query statement.

Index

cache_clear, [2](#)
cbs_add_date_column, [2](#)
cbs_add_label_columns, [3](#)
cbs_default_selection, [4](#)
cbs_download_data, [4](#), [6](#), [8](#)
cbs_download_meta, [5](#)
cbs_download_table, [5](#), [8](#)
cbs_extract_table_id, [6](#)
cbs_get_data, [3](#), [7](#), [8](#)
cbs_get_data_from_link, [8](#)
cbs_get_meta, [8](#), [9](#)
cbs_get_meta_from_dir, [9](#)
cbs_get_tables_themes, [10](#)
cbs_get_themes, [10](#)
cbs_get_toc, [5-7](#), [9](#), [11](#)

Date, [2](#)
download_data, [14](#)
download_data
 (download_data-deprecated), [12](#)
download_data-deprecated, [12](#)
download_meta
 (download_meta-deprecated), [13](#)
download_meta-deprecated, [13](#)
download_table, [13](#), [15](#)

get_data, [14](#)
get_meta, [15](#)
get_meta_from_dir, [16](#)
get_table_list, [14](#), [15](#), [17](#)
get_tables_themes, [16](#)
get_themes, [18](#)

resolve_deeplink, [19](#)