Package 'cbsodataR'

August 20, 2019

Type Package

Title Statistics Netherlands (CBS) Open Data API Client

Version 0.3.4

Description The data and meta data from Statistics Netherlands (<https://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

Encoding UTF-8

Imports whisker, jsonlite, utils

Suggests knitr, rmarkdown, dplyr, shiny

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Edwin de Jonge [aut, cre], Sara Houweling [ctb]

Maintainer Edwin de Jonge <edwindjonge@gmail.com>

Repository CRAN

Date/Publication 2019-08-20 12:30:02 UTC

R topics documented:

`cbsodataR-package` *Download all data from Statistics Netherlands / CBS*

Description

`cbsodataR` allows to download all official statistics of Statistics Netherlands (CBS) into R. For a introduction please visit the vignette: `vignette("cbsodataR",package="cbsodataR")`. The functions `cbs_get_toc()` and `cbs_get_data()` should get you going.

Catalog function

- `cbs_get_toc()`, returns a data.frame with table of contents (toc): the publication meta data for all available tables

Data retrieval

- `cbs_get_data()`, returns the data of a specific opendata/StatLine table
- `cbs_download_table()`, saves the data (and metadata) as csv files into a directory

Meta data

- `cbs_get_meta()`, returns the meta data objects of a specific opendata / StatLine table .
- `cbs_add_date_column()`, converts date/period codes into `DateTime` objects in the data set that was downloaded.
- `cbs_add_label_columns()`, adds labels to the code columns in the data that was downloaded.

Author(s)

Maintainer: Edwin de Jonge <edwindjonge@gmail.com>

Other contributors:

- Sara Houweling [contributor]

See Also

Useful links:

- https://github.com/edwindj/cbsodataR
- Report bugs at https://github.com/edwindj/cbsodataR/issues

---

cache_clear *clears the cache*

Description

clears the cache

Usage

```
cache_clear()
```

---

cbs_add_date_column *Convert the time variable into either a date or numeric.*

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

See Also

Other data retrieval: cbs_add_label_columns, cbs_download_data, cbs_extract_table_id, cbs_get_data_from_link, cbs_get_data

Other meta data: cbs_add_label_columns, cbs_download_meta, cbs_get_meta

---

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().

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.

Details

Code columns will be translated into label columns for each of the column that was supplied.

By default all code columns will be accompagnied with a label column. The name of each label column will be <code_column>_label.

Value

the original data.frame x with extra label columns. (see description)

See Also

Other data retrieval: cbs_add_date_column, cbs_download_data, cbs_extract_table_id, cbs_get_data_from_link, cbs_get_data

Other meta data: cbs_add_date_column, cbs_download_meta, cbs_get_meta

Examples

```
## Not run:

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

`x`	meta object
`...`	for future use

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

Description

Gets all data via bulk download. `cbs_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

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.

See Also

Other download: cbs_download_meta, cbs_download_table

Other data retrieval: cbs_add_date_column, cbs_add_label_columns, cbs_extract_table_id, cbs_get_data_from_link, cbs_get_data

---

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

See Also

Other meta data: cbs_add_date_column, cbs_add_label_columns, cbs_get_meta

Other download: cbs_download_data, cbs_download_table

`cbs_download_table` *Download a table from statistics Netherlands*

Description

`cbs_download_table` downloads the data and metadata of a table from statistics Netherlands and stores it in `csv` format.

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.

Details

`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.

Value

meta data object of `id` `cbs_get_meta()`.

See Also

Other download: cbs_download_data, cbs_download_meta

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.

See Also

Other data retrieval: `cbs_add_date_column`, `cbs_add_label_columns`, `cbs_download_data`, `cbs_get_data_from_link`, `cbs_get_data`

`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`.

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

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

Details

To reduce the download time, optionaly the data can be filtered on category values: for large tables (> 100k records) this is a wise thing to do.

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.

By default the columns will be converted to their type (typed=TRUE). CBS uses multiple types of missing (unknown, surpressed, 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()

Other data retrieval: cbs_add_date_column, cbs_add_label_columns, cbs_download_data, cbs_extract_table_id, cbs_get_data_from_link

Examples

```
## Not run:

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

## End(Not run)
```

`cbs_get_data_from_link` *Retrieve data from a link created from the StatLine app.*

Description

Retrieve data from a link created from the StatLine app.

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`

See Also

Other data retrieval: `cbs_add_date_column`, `cbs_add_label_columns`, `cbs_download_data`, `cbs_extract_table_id`, `cbs_get_data`

cbs_get_meta *Get metadata of a cbs table*

Description

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

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.

Details

The meta data of a CBS table is determined by the web api of Statistics Netherlands. cbsodataR stays close to this API. Each cbsodataR object has the following metadata items, which are all data.frames :

- $TableInfos: data.frame with the descriptive publication metadata of the table, such as Title, Description, Summary etc.
- $DataProperties: data.frame with the Title, Description, Unit etc. of each column in the dataset that is downloaded with cbs_get_data().
- $CategoryGroups: hierarchical groupings of the code columns.
- $<code column>: for each code column a data.frame with the Title, Key, Description etc. of each code / category in that column. e.g. Perioden for time codes c("2019JJ00","2018JJ00").

Value

cbs_table object containing several data.frames with meta data (see details)

See Also

Other meta data: cbs_add_date_column, cbs_add_label_columns, cbs_download_meta

cbs_get_meta_from_dir *Load meta data from a downloaded table*

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

...	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
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 protocal.

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

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

...	filter statement to select rows, e.g. Language="nl"
convert_dates	convert the columns with date-time information into DateTime (default TRUE)
select	character columns to be returned, by default all columns will be returned.
verbose	logical prints the calls to the webservice
cache	logical should the result be cached?
base_url	optionally specify a different server. Useful for third party data services implementing the same protocal.
include_ID	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

This method is deprecated in favor of cbs_download_data().

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.

See Also

Other download: cbs_download_meta, cbs_download_table

Other data retrieval: cbs_add_date_column, cbs_add_label_columns, cbs_extract_table_id, cbs_get_data_from_link, cbs_get_data

download_meta-deprecated

Dumps the meta data into a directory

Description

This method is deprecated in favor of cbs_download_meta().

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 (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

See Also

Other meta data: cbs_add_date_column, cbs_add_label_columns, cbs_get_meta

Other download: cbs_download_data, cbs_download_table

---

download_table-deprecated

Download a table from statistics Netherlands

Description

This method is deprecated in favor of cbs_download_table().

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 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.

Details

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.

Value

meta data object of id cbs_get_meta().

See Also

Other download: cbs_download_data, cbs_download_meta

Examples

```
## Not run:

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

## End(Not run)
```

---

get_data-deprecated *Get data from Statistics Netherlands (CBS)*

Description

This method is deprecated in favor of cbs_get_data()

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 cbs_get_toc()
...	optional filter statements, see details.
recode	recodes all codes in the code columns with their Title as found in the metadata
use_column_title	not used.
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.

Details

To reduce the download time, optionaly the data can be filtered on category values: for large tables (> 100k records) this is a wise thing to do.

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.

By default the columns will be converted to their type (`typed=TRUE`). CBS uses multiple types of missing (unknown, surpressed, 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`.

See Also

`cbs_get_meta()`, `cbs_download_data()`

Other data retrieval: `cbs_add_date_column`, `cbs_add_label_columns`, `cbs_download_data`, `cbs_extract_table_id`, `cbs_get_data_from_link`

Examples

```
## Not run:

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

## End(Not run)
```

---

get_meta-deprecated *Get meta data from table*

Description

This method is deprecated in favor of `cbs_get_meta()`

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 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.

Details

The meta data of a CBS table is determined by the web api of Statistics Netherlands. cbsodataR stays close to this API. Each cbsodataR object has the following metadata items, which are all data.frames :

- $TableInfos: data.frame with the descriptive publication metadata of the table, such as Title, Description, Summary etc.
- $DataProperties: data.frame with the Title, Description, Unit etc. of each column in the dataset that is downloaded with cbs_get_data().
- $CategoryGroups: hierarchical groupings of the code columns.
- $<code column>: for each code column a data.frame with the Title, Key, Description etc. of each code / category in that column. e.g. Perioden for time codes c("2019JJ00","2018JJ00").

Value

cbs_table object containing several data.frames with meta data (see details)

See Also

Other meta data: cbs_add_date_column, cbs_add_label_columns, cbs_download_meta

---

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 protocal.

Value

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

get_table_list *Retrieve a data.frame with requested cbs tables*

Description

This method is deprecated in favor of `cbs_get_toc()`.

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 protocal.

Value

`data.frame` with identifiers, titles and descriptions of tables

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` *Get list of all cbs thematic entries.*

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. `get_themes(ID=10)`.
`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 protocal.

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` *resolve a deeplink created in the opendata portal*

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