Package 'restatapi'

February 4, 2026

Type Package

Title Search and Retrieve Data from Eurostat Database

Date 2026-01-30

Version 0.24.5

Encoding UTF-8

Description Eurostat is the statistical office of the European Union and provides high quality statistics for Europe.
Large set of the data is disseminated through the Eurostat database (<https://ec.europa.eu/eurostat/web/main/data/database>).
The tools are using the REST API with the Statistical Data and Metadata eXchange (SDMX) Web Services
(<https://ec.europa.eu/eurostat/web/user-guides/data-browser/api-data-access/api-detailed-guidelines/sdmx2-1>) to search and download data from the Eurostat database using the SDMX standard.

License EUPL

Imports data.table, rjson, xml2

Suggests chron, knitr, rmarkdown, tinytest, remotes

NeedsCompilation no

URL https://github.com/eurostat/restatapi

BugReports https://github.com/eurostat/restatapi/issues

RoxygenNote 7.3.2

Author Mátyás Mészáros [aut, cre],
Sebastian Weinand [ctb]

Maintainer Mátyás Mészáros <matyas.meszaros@ec.europa.eu>

Repository CRAN

Date/Publication 2026-02-04 12:10:02 UTC

Contents

`.restatapi_env` *Create the cache environment*

Description

Create the cache environment

Usage

```
.restatapi_env
```

Format

An object of class `environment` of length 4.

clean_restatapi_cache *Clean restatapi cache*

Description

Remove all objects from the `.restatapi_env` except the configuration file, API version number, download method and the country codes. In addition, it deletes all the .rds files from the default and selected cache directory. See `get_eurostat_data` for more on cache.

Usage

```
clean_restatapi_cache(cache_dir = NULL, verbose = FALSE)
```

Arguments

`cache_dir`	a path to cache directory. If `NULL` (default) it will clean default temporary cache directory (`file.path(tempdir(),"restatapi")`). The default cache directory is used when the provided `cache_dir` does not exist. Directory can also be set with `options(restatapi_cache_dir=...)`.
`verbose`	a logical value with default `FALSE`, so detailed messages (for debugging) will not be printed. Can be set also with `options(restatapi_verbose=TRUE)`

Examples

```
clean_restatapi_cache(verbose=TRUE)
```

create_filter_table *Create a filter table*

Description

Create filter table from the `filters` and `date_filter` strings parameters of the `get_eurostat_data` to be used in the `filter_raw_data` function for filtering by query or on the local computer.

Usage

```
create_filter_table(
  filters,
  date_filter = FALSE,
  dsd = NULL,
  exact_match = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

filters	a string, a character or numeric vector or a named list containing words to filter by the different concepts, geographical location or time values. The words can be any word, Eurostat variable code, or value which are in the Data Structure Definition (DSD) and can be retrieved by the search_eurostat_dsd function. If a named list is used, then the name of the list elements should be the concepts from the DSD and the provided values will be used to filter the dataset for the given concept. The default is NULL, in this case no filter table is created. To filter by time see date_filter below. In case for filtering for time values, the date shall be defined as character string, and it should follow the format yyyy[-mm][-dd], where the month and the day part is optional.
date_filter	a logical value. If TRUE the filter table is genrated only for the time dimension. The default is FALSE, in this case a (dsd) should be provided which will be searched for the values given in the filters.
dsd	a table containing a DSD of an Eurostat dataset which can be retreived by the get_eurostat_dsd function.
exact_match	a logical value with the default value TRUE, if the strings provided in filters shall be matched exactly as it is or as a pattern in the DSD.
verbose	a logical value with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)
...	further arguments to the for search_eurostat_dsd function, e.g.: ignore.case or name. The ignore.case has the default value FALSE, then the strings provided in filters are matched as is, otherwise the case of the letters is ignored. If the name=FALSE then the pattern(s) provided in the filters argument is only searched in the code column of the DSD, and the names of the codes will not be searched.

Details

It is a sub-function to use in the get_eurostat_data to generate url for the given filters and date_filter in that function. The output can be used also for filtering data on the local computer with the get_eurostat_raw and filter_raw_data function, if the direct response from REST API did not provide data because of too large data set.

Value

a data.table containing in each row a distinct filtering condition to be applied to a raw Eurostat datatable or generate specific query.

If date_filter=TRUE, the output data table contains two columns with the following names:

sd	Starting date to be included in the filtered dataset, where date is formatted yyyy[-mm][-dd]
ed	End date of the period to be included in the filtered dataset, where the date is formatted yyyy[-mm][-dd]

In case date_filter=FALSE, the output tables have the following four columns:

pattern	Containing those parts of the filters string where the string part (pattern) was found in the dsd

concept	The name of the concepts corresponding to the result in the code/name column where the pattern was found in the
code	The list of codes where the pattern was found, or the code of a name (description of the code) where the pattern ap
name	The name (description of the code) which can be used as label for the code where the pattern was found, or the nar

See Also

get_eurostat_raw, search_eurostat_dsd, get_eurostat_data, filter_raw_data

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
dsd<-get_eurostat_dsd("avia_par_me")
create_filter_table(c("KYIV","hu","Quarterly"),dsd=dsd,exact_match=FALSE,ignore.case=TRUE)
create_filter_table(c("KYIV","LHBP","Monthly"),dsd=dsd,exact_match=FALSE,name=FALSE)
create_filter_table(c("2017-03",
                      "2001-03:2005",
                      "<2000-07-01",
                      2012:2014,
                      "2018<",
                      20912,
                      "<3452<",
                      ":2018-04>",
                      "2<034v",
                      "2008:2013"),
                    date_filter=TRUE,
                    verbose=TRUE)
options(timeout=60)
```

---

extract_data *Extract data values from SDMX XML*

Description

Extracts the data values from the SDMX XML data file

Usage

```
extract_data(
  xml_lf,
  keep_flags = FALSE,
  stringsAsFactors = FALSE,
  bulk = TRUE,
  check_toc = FALSE
)
```

Arguments

xml_lf	an input XML leaf with data series from an SDMX XML file to extract the value and its dimensions from it
keep_flags	a logical value if to extract the observation status (flag) information from the XML file. The default value is FALSE
stringsAsFactors	a logical value. If TRUE the columns are converted to factors. The default is FALSE, in this case the strings are returned as characters.
bulk	a logical value with default value TRUE if the input SDMX XML file is from the bulk download facility containing all the observations. If the input file has pre-filtered values then the value FALSE should be used.
check_toc	if the data file was downloaded using the URL from the TOC or not. The default is FALSE means not the TOC link is used.

Details

It is a sub-function to use in the get_eurostat_data and the get_eurostat_raw functions.

Value

a data frame containing the values of an SDMX node: the dimensions, value and the optional flag(s)

Examples

```
id<-"agr_r_milkpr"
url<-paste0("https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/data/",
            id,
            "?format=sdmx_2.1_structured&compressed=true")
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
  sdmx_xml<-get_compressed_sdmx(url,verbose=TRUE)
  xml_leafs<-xml2::xml_find_all(sdmx_xml,".//Series")
  extract_data(xml_leafs[1])
options(timeout=60)
```

---

extract_dsd *Extract the Data Structure Definition content from SDMX XML*

Description

Extracts values from the XML Data Structure Definition (DSD) file

Usage

```
extract_dsd(concept = NULL, dsd_xml = NULL, lang = "en")
```

Arguments

concept	a character vector with a concept id
dsd_xml	an XML file with DSD content
lang	a character string either en, de or fr to define the language version for the name column of the DSD. It is used only in the new API. The default is en - English.

Details

It is a sub-function to use in the get_eurostat_dsd function.

Value

a matrix with 3 columns if the provided concept has a code list in the DSD file. The first column is the provided concept. The second column is the possible codes under the given concept. The last column is the name/description for the code in the second column, which can be used as labels.

Examples

```
id<-"med_rd6"
cfg<-get("cfg",envir=restatapi:::.restatapi_env)
rav<-get("rav",envir=restatapi:::.restatapi_env)
dsd_url <- paste0(eval(
   parse(text=paste0("cfg$QUERY_BASE_URL$'",rav,"'$ESTAT$metadata$'2.1'$datastructure"))
  ),"/",eval(
    parse(text=paste0("cfg$QUERY_PRIOR_ID$'",rav,"'$ESTAT$metadata"))
  ),id,"?",eval(
    parse(text=paste0("cfg$QUERY_PARAMETERS$'",rav,"'$metadata[2]"))
  ),"=",eval(
    parse(text=paste0("cfg$DATAFLOW_REFERENCES$'",rav,"'$datastructure[1]"))
  )
 )
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
tryCatch({
  dsd_xml<-xml2::read_xml(dsd_url)},
  error=function(e){
  message("Unable to download the xml file.\n",e)},
  warning=function(w){
  message("Unable to download the xml file.\n",w)})
if (exists("dsd_xml")) extract_dsd("FREQ",dsd_xml)
options(timeout=60)
```

---

extract_toc *Extract the text of the table of contents from SDMX XML*

Description

Extracts the values of a node from the Eurostat XML Table of contents (TOC) file

Usage

```
extract_toc(ns)
```

Arguments

ns	an XML node set from the XML TOC file

Details

It is a sub-function to use in the get_eurostat_toc function.

Value

a character vector with all the values of the node set.

Examples

```
cfg<-get("cfg",envir=restatapi:::.restatapi_env)
rav<-get("rav",envir=restatapi:::.restatapi_env)
toc_endpoint<-eval(parse(text=paste0("cfg$TOC_ENDPOINT$'",rav,"'$ESTAT$xml")))

if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
tryCatch(xml_leafs<-xml2::xml_find_all(xml2::read_xml(toc_endpoint),".//nt:leaf"),
         error = function(e) {xml_leafs<-""},
         warning = function(w) {xml_leafs<-""})
if (exists("xml_leafs")){
   if (Sys.info()[['sysname']]=='Windows'){
      xml_node<-as.character(xml_leafs[1])
   }else{
      xml_node<-xml_leafs[1]
   }
   restatapi::extract_toc(xml_node)
}
options(timeout=60)
```

---

filter_raw_data *Filter raw data locally*

Description

Filter downloaded full raw dataset on local computer if the get_eurostat_data has not provided data due to too large datasets for the REST API.

Usage

```
filter_raw_data(raw_data = NULL, filter_table = NULL, date_filter = FALSE)
```

Arguments

raw_data	an input data.table dataset resulted from the call of the get_eurostat_raw function
filter_table	a data table with values for the concepts or time to be filtered out which can be generated by the create_filter_table function
date_filter	a logical value. If TRUE the filter table should be applied to the time columns of the raw_data. The default is FALSE, in this case the filters applied to the other columns of the raw_data.

Details

It is a sub-function to use in the get_eurostat_data to filter data on the local computer if the direct response from REST API did not provide data because of too large data set (more than 30 thousands observations). The filter_table contains always at least two columns. In case if date_filter=TRUE then the two columns should have the following names and the provided conditions are applied to the time column of the the raw_data data.table.

sd	Starting date to be included, where date is formatted as yyyy[-mm][-dd] (the month and day are optional)
ed	End date of the period to be included in the dataset formatted as yyyy[-mm][-dd] (the month and day are optional)

In case if date_filter=FALSE then the columns should have the following names:

concept	Containing concept names, which is a column name in the raw_data data.table
code	A possible code under the given concept, which is a value in the column of the raw_data data.table defined by the c

Value

a filtered data.table containing only the rows of raw_data which fulfills the conditions in the filter_table

See Also

get_eurostat_raw, search_eurostat_dsd, get_eurostat_data, create_filter_table

Examples

```
id<-"tus_00age"
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
rd<-get_eurostat_raw(id)
dsd<-get_eurostat_dsd(id)
ft<-create_filter_table(c("TIME_SP","Hungary",'T'),FALSE,dsd)
filter_raw_data(rd,ft)
options(timeout=60)
```

`get_compressed_sdmx` *Download and extract compressed SDMX XML*

Description

Downloads and extracts the data values from the SDMX XML data file

Usage

```
get_compressed_sdmx(url = NULL, verbose = FALSE, format = "gz")
```

Arguments

`url`	a URL from the bulk download facility to download the zipped SDMX XML file
`verbose`	a logical value with default `FALSE`, so detailed messages (for debugging) will not printed. Can be set also with `options(restatapi_verbose=TRUE)`.
`format`	the format of the compression, either "zip" or "gz" the default value

Details

It is a sub-function to use in the `get_eurostat_raw` and the `get_eurostat_data` functions.

Value

an xml class object with SDMX tags extracted and read from the downloaded file.

Examples

```
id<-"agr_r_milkpr"
url<-paste0("https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/data/",
            id,
            "?format=sdmx_2.1_structured&compressed=true")
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
sdmx_xml<-get_compressed_sdmx(url,verbose=TRUE,format="gz")
options(timeout=60)
```

get_eurostat_bulk *Get Eurostat data in a standardized format*

Description

Download data sets from Eurostat database and put in a standardized format.

Usage

```
get_eurostat_bulk(
  id,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = TRUE,
  select_freq = NULL,
  keep_flags = FALSE,
  cflags = FALSE,
  check_toc = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

id	a code name for the dataset of interest. See search_eurostat_toc for details how to get an id.
cache	a logical value whether to do caching. Default is TRUE.
update_cache	a logical value with a default value FALSE, whether to update cache. Can be set also with options(restatapi_update=TRUE).
cache_dir	a path to a cache directory. The NULL (default) uses the memory as cache. If the folder cache_dir directory does not exist it saves in the 'restatapi' directory under the temporary directory from tempdir(). Directory can also be set with option(restatapi_cache_dir=...).
compress_file	a logical value whether to compress the RDS-file in caching. Default is TRUE.
stringsAsFactors	a logical value with the default TRUE. In this case the columns are converted to factors. If FALSE, the strings are returned as characters.
select_freq	a character symbol for a time frequency when a dataset has multiple time frequencies. Possible values are: A = annual, S = semi-annual, H = half-year, Q = quarterly, M = monthly, W = weekly, D = daily. The default is NULL as most datasets have only one time frequency. In case if there are multiple frequencies and select_freq=NULL, then only the most common frequency kept. If all the frequencies needed the get_eurostat_raw function can be used.

keep_flags	a logical value whether the observation status (flags) - e.g. "confidential", "provisional", etc. - should be kept in a separate column or if they can be removed. Default is FALSE. For flag values see: https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/codelist/ESTAT/OBS_STATUS/?compressed=false&format=TSV&lang=en.
cflags	a logical value whether the missing observations with flag 'c' - "confidential" should be kept or not. Default is FALSE, in this case these observations dropped from the dataset. If this parameter TRUE then all the flags and the suppressed observations with missing values are kept. In this case the parameter provided in keep_flags is set to TRUE.
check_toc	a logical value whether to check the provided id in the Table of Contents (TOC) or not. The default value FALSE, in this case the base URL for the download link is retrieved from the configuration file. If the value is TRUE then the TOC is downloaded and the id is checked in it. If it found there then the download link is retrieved form the TOC.
verbose	a logical value with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE).
...	other parameter(s) to pass on the load_cfg function

Details

Data sets are downloaded from the Eurostat bulk download facility in TSV format as in this case smaller file has to be downloaded and processed. If there is more then one frequency then the dataset is filtered for a unique time frequency. If no frequency is selected and there are multiple frequencies in the dataset, then the most common value is used used for frequency.

Compared to the ouptut of the get_eurostat_raw function, the frequency (FREQ) and time format (TIME_FORMAT) columns are not included in the bulk data and the column names for the time period, observation values and status have standardised names: "time", "values" and "flags" independently if the data was downloaded previously in SDMX or TSV format.

By default all datasets cached as they are often rather large. The datasets cached in memory (default) or can be stored in a temporary directory if cache_dir or option(restatpi_cache_dir) is defined. The cache can be emptied with clean_restatapi_cache.

The id, is a value from the code column of the table of contents (get_eurostat_toc), and can be searched for it with the search_eurostat_toc function. The id value can be retrieved from the Eurostat database as well. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis.

Value

a data.table with the following columns:

dimension names	One column for each dimension in the data
time	A column for the time dimension
values	A column for numerical values
flags	A column for flags if the keep_flags=TRUE or cflags=TRUE otherwise this column is not included in the c

The data.table does not include all missing values. The missing values are dropped if both the value and the flag is missing on a particular time.

See Also

get_eurostat_data, get_eurostat_raw

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
head(get_eurostat_bulk("agr_r_milkpr",keep_flags=TRUE))
options(restatapi_update=TRUE)
head(get_eurostat_bulk("avia_par_ee",check_toc=TRUE))
head(get_eurostat_bulk("avia_par_ee",select_freq="A",verbose=TRUE))
options(restatapi_update=FALSE)
head(get_eurostat_bulk("agr_r_milkpr",cache_dir=tempdir(),compress_file=FALSE,verbose=TRUE))
clean_restatapi_cache(cache_dir=tempdir(),verbose=TRUE)
options(timeout=60)
```

---

get_eurostat_cache *Load an object from cache*

Description

Search and load the object (dataset/toc/DSD) from cache

Usage

```
get_eurostat_cache(oname, cache_dir = NULL, verbose = FALSE)
```

Arguments

oname	a character string with the name of the object (toc, dataset id, DSD id)
cache_dir	a path to a cache directory to search in. The default is NULL, in this case the object is searched in the memory (in the .restatapi_env). Otherwise if the cache_dir directory does not exist it searches the 'restatapi' directory in the temporary directory from tempdir(). Directory can also be set with options(restatapi_cache_dir=...).
verbose	a logical value with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE).

Details

If the given name or the beginning of the name (for datasets) found in the cache then it returns the value of the object otherwise it returns NULL.

Value

the requested object if exists in the '.restatapi_env' or in the cache_dir, otherwise it returns the NULL value.

Examples

```
dt<-data.frame(txt=c("a","b","c"),nr=c(1,2,3))
put_eurostat_cache(dt,"teszt")
get_eurostat_cache("teszt",verbose=TRUE)
```

`get_eurostat_codelist` *Download the codelist of a concept*

Description

Download codelist of a concept from Eurostat if it is not cached previously.

Usage

```
get_eurostat_codelist(
  id,
  lang = "en",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

`id`	a character string with id of the concept. It is a value from the `concept` column of the `get_eurostat_dsd` function.
`lang`	a character string either `en`, `de` or `fr` to define the language version for the name column of the codelist. It is used only in the new API. The default is `en` - English.
`cache`	a boolean whether to load/save the TOC from/in the cache or not. The default value is `TRUE`, so that the TOC is checked first in the cache and if does not exist then downloaded from Eurostat and cached.
`update_cache`	a boolean to update cache or not. The default value is `FALSE`, so the cache is not updated. Can be set also with `options(restatapi_update=TRUE)`
`cache_dir`	a path to a cache directory. The default is `NULL`, in this case the TOC is cached in the memory (in the '.restatapi_env'). Otherwise if the `cache_dir` directory does not exist it creates the 'restatapi' directory in the temporary directory from `tempdir()` to save the RDS-file. Directory can also be set with `option(restatapi_cache_dir=...)`.
`compress_file`	a logical whether to compress the RDS-file in caching. Default is `TRUE`.

verbose	A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)
...	parameter to pass on the load_cfg function

Details

The codelist is downloaded from Eurostat's website, through the REST API in XML (SDMX-ML) format.

Value

If the codelist does not exist it returns NULL otherwise the result is a table with the 2 columns:

code	All the possible codes under the concept
name	The name/description of the code

References

For more information see the detailed documentation of the API.

See Also

get_eurostat_dsd.

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
get_eurostat_codelist("freq",lang="de",cache=FALSE,verbose=TRUE)
options(timeout=60)
```

---

get_eurostat_data *Download, extract and filter Eurostat data*

Description

Download full or partial data set from Eurostat database.

Usage

```
get_eurostat_data(
  id,
  filters = NULL,
  lang = "en",
  exact_match = TRUE,
  date_filter = NULL,
  label = FALSE,
  select_freq = NULL,
  cache = TRUE,
```

```
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = TRUE,
  keep_flags = FALSE,
  cflags = FALSE,
  check_toc = FALSE,
  local_filter = TRUE,
  force_local_filter = FALSE,
  mode = "xml",
  verbose = FALSE,
  ...
)
```

Arguments

id	A code name for the dataset of interest. See search_eurostat_toc for details how to get an id.
filters	a string, a character vector or named list containing words to filter by the different concepts or geographical location. If filter applied only part of the dataset is downloaded through the API. The words can be any word, Eurostat variable code, and value which are in the DSD search_eurostat_dsd. If a named list is used, then the name of the list elements should be the concepts from the DSD and the provided values will be used to filter the dataset for the given concept. The default is NULL, in this case the whole dataset is returned via the bulk download. To filter by time see date_filter below. If after filtering still the dataset has more observations than the limit per query via the API, then the raw download is used to retrieve the whole dataset and apply the filter on the local computer. This option can be disabled with the local_filter=FALSE parameter.
lang	a character string either en, de or fr to define the language version for the DSD to search in for the filters. The default is en - English.
exact_match	a boolean with the default value TRUE, if the strings provided in filters shall be matched exactly as it is or as a pattern.
date_filter	a vector which can be numeric or character containing dates to filter the dataset. If date is defined as character string it should follow the format yyyy[-mm][-dd], where the month and the day part is optional. If date filter applied only part of the dataset is downloaded through the API. The default is NULL, in this case the whole dataset is returned via the bulk download. If after filtering still the dataset has more observations than the limit per query via the API, then the raw download is used to retrieve the data and apply the filter on the local computer. This option can be disabled with the local_filter=FALSE parameter.
label	a boolean with the default FALSE. If it is TRUE then the code values are replaced by the name from the Data Structure Definition (DSD) get_eurostat_dsd. For example instead of "D1110A", "Raw cows' milk from farmtype" is used or "HU32" is replaced by "Észak-Alföld".

`select_freq` a character symbol for a time frequency when a dataset has multiple time frequencies. Possible values are: A = annual, S = semi-annual, H = half-year, Q = quarterly, M = monthly, W = weekly, D = daily. The default is `NULL` as most datasets have just one time frequency and in case there are multiple frequencies, then only the most common frequency kept. If all the frequencies needed the `get_eurostat_raw` can be used.

`cache` a logical whether to do caching. Default is `TRUE`. Affects only queries without filtering. If `filters` or `date_filter` is used then there is no caching.

`update_cache` a logical with a default value `FALSE`, whether to update the data in the cache. Can be set also with `options(restatapi_update=TRUE)`

`cache_dir` a path to a cache directory. The `NULL` (default) uses the memory as cache. If the folder `cache_dir` directory does not exist it saves in the 'restatapi' directory under the temporary directory from `tempdir()`. Directory can also be set with `option(restatapi_cache_dir=...)`.

`compress_file` a logical whether to compress the RDS-file in caching. Default is `TRUE`.

`stringsAsFactors` if `TRUE` (the default) the non-numeric columns are converted to factors. If the value `FALSE` they are returned as characters.

`keep_flags` a logical whether the observation status (flags) - e.g. "confidential", "provisional", etc. - should be kept in a separate column or if they can be removed. Default is `FALSE`. For flag values see: https://ec.europa.eu/eurostat/api/dissemination/sdmx/2.1/codelist/ESTAT/OBS_STATUS/?compressed=false&format=TSV&lang=en.

`cflags` a logical whether the missing observations with flag 'c' - "confidential" should be kept or not. Default is `FALSE`, in this case these observations dropped from the dataset. If this parameter `TRUE` then the flags are kept and the parameter provided in `keep_flags` is not taken into account.

`check_toc` a boolean whether to check the provided `id` in the Table of Contents (TOC) or not. The default value `FALSE`, in this case the base URL for the download link is retrieved from the configuration file. If the value is `TRUE` then the TOC is downloaded and the `id` is checked in it. If it found then the download link is retrieved form the TOC.

`local_filter` a boolean whether do the filtering on the local computer or not in case after filtering still the dataset has more observations than the limit per query via the API would allow to download. The default is `TRUE`, in this case if the response footer contains information that the result cannot be downloaded becuase it is too large, then the whole raw dataset is downloaded and filtered on the local computer.

`force_local_filter` a boolean with the default value `FALSE`. In case, if there are existing filter conditions, then it will do the filtering on the local computer and not requesting through the REST API. It can be useful, if the values are not numeric as these are provided as NaN (Not a Number) through the REST API, but it is fully listed in the raw dataset.

`mode` defines the format of the dataset response from the API. It can be `csv` for SDMX-CSV or `xml` for the SDMX-ML version.

verbose — A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)

... — further arguments to the for search_eurostat_dsd function, e.g.: ignore.case or name. The ignore.case has the default value FALSE, then the strings provided in filters are matched as is, otherwise the case of the letters is ignored. If the name=FALSE then the pattern(s) provided in the filters argument is only searched in the code column of the DSD, and the names of the codes will not be searched.

Details

Data sets are downloaded from the Eurostat Web Services SDMX API if there is a filter otherwise the the Eurostat bulk download facility is used. If only the table id is given, the whole table is downloaded from the bulk download facility. If also filters or date_filter is defined then the SDMX REST API is used. In case after filtering the dataset has more rows than the limitation of the SDMX REST API (1 million values at one time) then the bulk download is used to retrieve the whole dataset .

By default all datasets cached as they are often rather large. The datasets cached in memory (default) or can be stored in a temporary directory if cache_dir or option(restatpi_cache_dir) is defined. The cache can be emptied with clean_restatapi_cache.

The id, is a value from the code column of the table of contents (get_eurostat_toc), and can be searched for with the search_eurostat_toc function. The id value can be retrieved from the Eurostat database as well. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis.

Filtering can be done by the codes as described in the API documentation providing in the correct order and connecting with "." and "+". If we do not know the codes we can filter based on words or by the mix of the two putting in a vector like c("AT$","Belgium","persons","Total"). Be careful that the filter is case sensitive, if you do not know the code or label exactly you can use the option ignore.case=TRUE and exact_match=FALSE, but in this case the results may include unwanted elements as well. In the filters parameter regular expressions can be used as well. We do not have to worry about the correct order of the filter, it will be put in the correct place based on the DSD.

The date_filter shall be a string in the format yyyy[-mm][-dd]. The month and the day part is optional, but if we use the years and we have monthly frequency then all the data for the given year is retrieved. The string can be extended by adding the "<" or ">" to the beginning or to the end of the string. In this case the date filter is treated as range, and the date is used as a starting or end date. The data will include the observation of the start/end date. A single date range can be defined as well by concatenating two dates with the ":", e.g. "2016-08:2017-03-15". As seen in the example the dates can have different length: one defined only at year/month level, the other by day level. If a date range is defined with ":", it is not possible to use the "<" or ">" characters in the date filter. If there are multiple dates which is not a continuous range, it can be put in vector in any order like c("2016-08",2013:2015,"2017-07-01"). In this case, as well, it is not possible to use the "<" or ">" characters.

Value

a data.table with the following columns:

freq	A column for the frequency of the data in case there are multiple frequencies, for single frequency this col
dimension names	One column for each dimension in the data
time	A column for the time dimension
values	A column for numerical values
flags	A column for flags if the keep_flags=TRUE or cflags=TRUE otherwise this column is not included in the

The data.table does not include all missing values. The missing values are dropped if the value and flag are missing on a particular time.

In case the provided filters can be found in the DSD, then it is used to query the API or applied locally. If the applied filters with combination of date_filter and select_freq has no observation in the data set then the fucntion returns the data.table with 0 row.

In case none of the provided filters, date_filter or select_freq can be parsed or found in the DSD then the whole dataset downloaded through the bulk download with a warning message.

In case the id is not exist then the function returns the value NULL.

See Also

search_eurostat_toc, search_eurostat_dsd, get_eurostat_bulk

Examples

```
load_cfg()
eu<-get("cc",envir=restatapi::.restatapi_env)



if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
head(get_eurostat_data("NAMA_10_GDP"))
head(get_eurostat_data("htec_cis3",update_cache=TRUE,check_toc=TRUE,verbose=TRUE))
head(get_eurostat_data("agr_r_milkpr",cache_dir="/tmp",cflags=TRUE))
options(restatapi_update=FALSE)
options(restatapi_cache_dir=file.path(tempdir(),"restatapi"))
head(get_eurostat_data("avia_gonc",select_freq="A",cache=FALSE))
head(get_eurostat_data("agr_r_milkpr",date_filter=2008,keep_flags=TRUE))
dt<-get_eurostat_data("avia_par_me",
                      filters="BE$",
                      exact_match=FALSE,
                      date_filter=c(2016,"2017-03","2017-07-01"),
                      select_freq="Q",
                      label=TRUE,
                      name=FALSE)
dt<-get_eurostat_data("agr_r_milkpr",
                      filters=c("BE$","Ungarn"),
                      lang="de",
                      date_filter="2007-06<",
                      keep_flags=TRUE)
dt<-get_eurostat_data("nama_10_a10_e",
                    filters=c("Annual","EU28","Belgium","AT","Total","EMP_DC","person"),
                      date_filter=c("2008",2002,2013:2018))
dt<-get_eurostat_data("vit_t3",
```

```
                      filters=c("EU28",eu$EA15,"HU$"),
                      date_filter=c("2015",2007))
dt<-get_eurostat_data("avia_par_me",
                      filters="Q...ME_LYPG_HU_LHBP+ME_LYTV_UA_UKKK",
                      date_filter=c("2016-08","2017-07-01"),
                      select_freq="M")
dt<-get_eurostat_data("htec_cis3",
                       filters="lu",
                       ignore.case=TRUE)
dt<-get_eurostat_data("bop_its6_det",
                       filters=list(bop_item="SC",
                                    currency="MIO_EUR",
                                    partner="EXT_EU28",
                                    geo=c("EU28","HU"),
                                    stk_flow="BAL",
                                    time="2015:2020"),
                       date_filter="2010:2012",
                       select_freq="A",
                       label=TRUE,
                       name=FALSE)
clean_restatapi_cache("/tmp",verbose=TRUE)
options(timeout=60)
```

`get_eurostat_dsd` *Download the Data Structure Definition of a dataset*

Description

Download Data Structure Definition (DSD) of a Eurostat dataset if it is not cached previously.

Usage

```
get_eurostat_dsd(
  id,
  lang = "en",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

`id`	a character string with the id of the dataset. It is a value from the `codename` column of the `get_eurostat_toc` function.

lang	a character string either en, de or fr to define the language version for the name column of the DSD. It is used only in the new API. The default is en - English.
cache	a boolean whether to load/save the DSD from/in the cache or not. The default value is TRUE, so that the DSD is checked first in the cache and if does not exist then downloaded from Eurostat and cached.
update_cache	a boolean to update cache or not. The default value is FALSE, so the cache is not updated. Can be set also with options(restatapi_update=TRUE)
cache_dir	a path to a cache directory. The default is NULL, in this case the DSD is cached in the memory (in the '.restatapi_env'). Otherwise if the cache_dir directory does not exist it creates the 'restatapi' directory in the temporary directory from tempdir() to save the RDS-file. Directory can also be set with option(restatapi_cache_dir=...).
compress_file	a logical whether to compress the RDS-file in caching. Default is TRUE.
verbose	A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)
...	parameter to pass on the load_cfg function

Details

The DSD is downloaded from Eurostat's website, through the REST API in XML (SDMX-ML) format.

Value

If the DSD does not exist it returns NULL otherwise the result is a table with the 3 columns:

concept	The name of the concepts in the order of the data structure
code	The possible list of codes under the concept
name	The name/description of the code

References

For more information see the detailed documentation of the API.

See Also

get_eurostat_data, search_eurostat_toc.

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
head(get_eurostat_dsd("med_rd6",lang="de",cache=FALSE,verbose=TRUE))
options(timeout=60)
```

get_eurostat_raw *Get Eurostat data as it is*

Description

Download data sets from Eurostat database .

Usage

```
get_eurostat_raw(
  id,
  mode = "txt",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = FALSE,
  keep_flags = FALSE,
  check_toc = FALSE,
  melt = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

id	A code name for the dataset of interest. See search_eurostat_toc for details how to get an id.
mode	defines the format of the downloaded dataset. It can be txt (the default value) for Tab Separated Values (TSV), or csv for SDMX-CSV, or xml for the SDMX-ML version.
cache	a logical whether to do caching. Default is TRUE.
update_cache	a logical with a default value FALSE, whether to update cache. Can be set also with options(restatapi_update=TRUE)
cache_dir	a path to a cache directory. The NULL (default) uses the memory as cache. If the folder if the cache_dir directory does not exist it saves in the 'restatapi' directory under the temporary directory from tempdir(). Directory can also be set with option(restatapi_cache_dir=...).
compress_file	a logical whether to compress the RDS-file in caching. Default is TRUE.
stringsAsFactors	if TRUE the variables which are not numeric are converted to factors. The default value FALSE, in this case they are returned as characters.
keep_flags	a logical whether the observation status (flags) - e.g. "confidential", "provisional", etc. - should be kept in a separate column or if they can be removed. Default is FALSE. For flag values see: https://ec.europa.eu/eurostat/api/

dissemination/sdmx/2.1/codelist/ESTAT/OBS_STATUS/?compressed=false&format=TSV&lang=en.

check_toc — a boolean whether to check the provided id in the Table of Contents (TOC) or not. The default value FALSE, in this case the base URL for the download link is retrieved from the configuration file. If the value is TRUE then the TOC is downloaded and the id is checked in it. If it found then the download link is retrieved form the TOC.

melt — a boolean with default value TRUE and used only if the mode="txt". In case it is FALSE, the downloaded tsv file is not melted, the time dimension remains in columns and it does not process the flags.

verbose — A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)

... — further argument for the load_cfg function

Details

Data sets are downloaded from the Eurostat bulk download facility in CSV, TSV or SDMX format.

The id, should be a value from the code column of the table of contents (get_eurostat_toc), and can be searched for with the search_eurostat_toc function. The id value can be retrieved from the Eurostat database as well. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis. By default all datasets downloaded in TSV format and cached as they are often rather large. The datasets cached in memory (default) or can be stored in a temporary directory if cache_dir or option(restatpi_cache_dir) is defined. The cache can be emptied with clean_restatapi_cache. If the id is checked in TOC then the data will saved in the cache with the date from the "lastUpdate" column from the TOC, otherwise it is saved with the current date.

Value

a data.table with the following columns if the default melt=TRUE is used:

FREQ	The frequency of the data (**A**nnual, **S**emi-annual, **H**alf-year, **Q**uarterly, **M**onthly, **W**eekly, **D**aily)
dimension names	One column for each dimension in the data
TIME_FORMAT	A column for the time format, if the source file SDMX-ML and the data was not loaded from a previou
time/TIME_PERIOD	A column for the time dimension, where the name of the column depends on the source file (TSV/SDM
values/OBS_VALUE	A column for numerical values, where the name of the column depends on the source file (TSV/SDMX
flags/OBS_STATUS	A column for flags if the keep_flags=TRUE otherwise this column is not included in the data table, and

The data does not include all missing values. The missing values are dropped if the value and flags are missing on a particular time.

In case melt=FALSE the results is a data.table where the first column contains the comma separated values of the various dimensions, and the columns contains the observations for each time dimension.

See Also

get_eurostat_data, get_eurostat_bulk

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
head(get_eurostat_raw("agr_r_milkpr",keep_flags=TRUE))
head(get_eurostat_raw("avia_par_ee",mode="xml",check_toc=TRUE,update_cache=TRUE,verbose=TRUE))
options(restatapi_update=FALSE)
head(get_eurostat_raw("avia_par_me",mode="txt",melt=FALSE))
head(get_eurostat_raw("avia_par_me",
                      mode="txt",
                      cache_dir=tempdir(),
                      compress_file=FALSE,
                      verbose=TRUE))
options(timeout=60)
```

---

get_eurostat_toc *Download the Table of Contents of Eurostat datasets*

Description

Download Table of Contents (TOC) of Eurostat datasets if it is not cached previously.

Usage

```
get_eurostat_toc(
  mode = "xml",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  lang = "en",
  verbose = FALSE,
  ...
)
```

Arguments

mode	a character string either `xml` or `txt` defining the download mode. Depending on the mode the 'xml' version or the 'text' version of the TOC is downloaded. The deafult value is `xml` as it provides more information (e.g. number of values, short description and download links in different formats (SDMX, TSV))
cache	a boolean whether to load/save the TOC from/in the cache or not. The default value is `TRUE`, so that the TOC is checked first in the cache and if does not exist then downloaded from Eurostat and cached.
update_cache	a boolean to update cache or not. The default value is `FALSE`, so the cache is not updated. Can be set also with `options(restatapi_update=TRUE)`

cache_dir	a path to a cache directory. The default is NULL, in this case the TOC is cached in the memory (in the '.restatapi_env'). Otherwise if the cache_dir directory does not exist it creates the 'restatapi' directory in the temporary directory from tempdir() to save the RDS- file. Directory can also be set with option(restatapi_cache_dir=...).
compress_file	a logical whether to compress the RDS-file in caching. Default is TRUE.
lang	a character string either en, de or fr to define the language version for the table of contents. The default is en - English.
verbose	A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)
...	parameter to pass on the load_cfg function

Details

The TOC is downloaded from Eurostat websites through the REST API for the xml (default) version or from the bulk download facilities for txt version. From the downloaded TOC the values in the 'code' column can be used as id in the get_eurostat_dsd, get_eurostat_raw, get_eurostat_bulk, and get_eurostat_data functions.

Value

A data table with the following columns:

title	The name of dataset/table in the language provided by the lang parameter
code	The codename of dataset/table which can be used as id in other functions
type	The type of information: 'dataset' or 'table'
lastUpdate	The date when the data was last time updated for tables and datasets
lastModified	The date when the structure of the dataset/table was last time modified
dataStart	The start date of the data in the dataset/table
dataEnd	The end date of the data in the dataset/table
values	The number of values in the dataset/table, and it is filled only if the download mode is "xml"
unit	The unit name for tables in the language provided by the lang parameter, for dataset it is empty and thi
source	The source of the data and it is filled only if the download mode is "xml"
shortDescription	The short description of the values for tables in the language provided by the lang parameter, for datase
metadata.html	The link to the metadata in html format, and this column exists only if the download mode is "xml"
metadata.sdmx	The link to the metadata in SDMX format, and this column exists only if the download mode is "xml"
downloadLink.tsv	The link to the whole dataset/table in tab separated values format in the bulk download facility and this

References

For more technical information see the detailed documentation of the API.

See Also

search_eurostat_toc, get_eurostat_dsd, get_eurostat_raw, get_eurostat_bulk, get_eurostat_data.

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
toc_xml<-get_eurostat_toc(cache=FALSE,verbose=TRUE)
head(toc_xml)
toc_txt<-get_eurostat_toc(mode="txt", lang="de")
head(toc_txt)
options(timeout=60)
```

---

load_cfg *Load configuration data from JSON*

Description

Load the configuration information to the '.restatapi_env' from the JSON configuration file.

Usage

```
load_cfg(
  api_version = "default",
  cfg_file = "github",
  load_toc = FALSE,
  parallel = TRUE,
  max_cores = FALSE,
  verbose = FALSE
)
```

Arguments

api_version	It can be either "old", "new", "test" or "current". The default value is "current" which defined by the DEFAULT_API_VERSION value of the config file.
cfg_file	The location of the config file. It can be either "github" (the default value) or "local".
load_toc	The default value FALSE, which means that the XML version of the Table of contents (TOC) will not be downloaded and cached automatically in the '.restatapi_env' when the package is loaded.
parallel	A boolean with the default value TRUE. If there are multiple cores/logical processors then part of the data extraction is made in parallel reducing significantly the time needed for large datasets. If the value is FALSE the option restatapi_cores set to 1.
max_cores	A boolean with the default value FALSE. If the parameter 'parallel' is TRUE then this parameter is taken into account otherwise it is ignored. If the value is TRUE, then the maximum minus one cores/logical processors are used for parallel computing. If the parameter FALSE, then the default value of getOption("mc.cores") is used, if it is defined. If mc.cores is NULL then depending on the memory size

and number of available cores/threads the `restatapi_cores` are set to 2 or 4 cores/logical processors. Otherwise the parallel processing turned off by setting the option `restatapi_cores` to 1. The number of cores used for parallel computing can be changed any time with `options(restatapi_cores=...)`

`verbose` A boolean if the verbose message about the configuration to be showed or not. The default is `FALSE`. Can be set also with `options(restatapi_verbose=TRUE)`

Details

Loads configuration data from a JSON file. The function first tries to load the configuration file from GitHub. If it is not possible it loads from the file delivered with the package. By this way different version of the API can be tested. Since in many cases there is http/https redirection in the download which can cause problems with the 'wininet' download method, the 'libcurl' method is used when it is available. This configuration code sets up the parallel processing to handle large XML files efficiently. By default if there is more then 4 cores/logical processors and at least 32 GB of RAM then 4 cores are used for parallel computing. If there is more then 2 cores then 2 cores are used. This default configuration can be overwritten with `options(restatapi_cores=...)` or with the `max_cores=TRUE` parameter. In the second case part of the computation distributed over the maximum number minus one cores. By using the `max_cores=TRUE` option there is a higher probability that the program will run out off memory for larger datasets. In addition, the list of country codes are loaded to the variable `cc` (country codes), based on the Eurostat standard code list

Value

it returns 4 objects in the '.restatapi_env'

- `cfg` a list with all the configuration data
- `rav` a character string with a number defining the API_VERSION from the configuration file to be used later. It is determined based on the `api_version` parameter.
- `cc` a list containing the 2 character country codes of the member states for different EU composition like EU15, EU28 or EA (Euro Area).
- `dmethod` the download method to be used to access Eurostat database. If the 'libcurl' method exists under Windows then it will be the default method for file download, otherwise it will be set 'auto'. The download method can be changed any time with `options(restatapi_dmethod=...)`

Examples

```
load_cfg(parallel=FALSE)
options(restatapi_dmethod="auto")
load_cfg(api_version="test",verbose=TRUE,max_cores=FALSE)
load_cfg()
eu<-get("cc",envir=.restatapi_env)
eu$EU28
eu$EA15
```

`put_eurostat_cache` *Put an object to cache*

Description

Save the object (dataset/toc/DSD) to cache

Usage

```
put_eurostat_cache(
  obj,
  oname,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE
)
```

Arguments

`obj`	an object (toc, dataset, DSD)
`oname`	a character string with the name of the object to reference later in the cache
`update_cache`	a logical with a default value `FALSE`, whether to update the cache. In this case the existing value in the cache is overwritten. Can be set also with `options(restatapi_update=TRUE)`
`cache_dir`	a path to a cache directory. The default is `NULL`, in this case the object is saved in the memory (in the '.restatapi_env'). Otherwise if the `cache_dir` directory does not exist it saves in the 'restatapi' directory under the temporary directory from `tempdir()`. Directory can also be set with `options(restatapi_cache_dir=...)`.
`compress_file`	a logical whether to compress the RDS-file in caching. Default is `TRUE`.

Details

Saves a given object in cache. This can be the memory `.restatapi_env` or on the hards disk. If the given `cache_dir` does not exist then the file is saved in the R temp directory (`tempdir()`). If the file or object with the `oname` exists in the cache, then the object is not cached.

Value

The function returns the place where the object was cached: either it creates an the object in the memory ('.restatapi_env') or creates an RDS-file.

Examples

```
dt<-data.frame(txt=c("a","b","c"),nr=c(1,2,3))
put_eurostat_cache(dt,"teszt")
get("teszt",envir=restatapi::.restatapi_env)
put_eurostat_cache(dt,"teszt",cache_dir=tempdir())
readRDS(file.path(tempdir(),"teszt.rds"))
```

```
clean_restatapi_cache(cache_dir=tempdir())
```

`search_eurostat_dsd` *Search for pattern in the Data Structure Definition of a dataset*

Description

Search the Data Structure Definition (DSD) of a Eurostat dataset for a given pattern. It returns the rows where the pattern appears in the code and name column of the output of the `get_eurostat_dsd` function.

Usage

```
search_eurostat_dsd(pattern, dsd = NULL, name = TRUE, exact_match = FALSE, ...)
```

Arguments

`pattern`	a character string or a vector of character string.
`dsd`	a table containing Data Structure Definition (DSD) of a Eurostat dataset which can be retreived by the `get_eurostat_dsd` function.
`name`	a boolean with the default value `TRUE`, if the search shall look for the pattern in the name of the code. If the value `FALSE`, then only the 'code' column of the DSD will be searched.
`exact_match`	a boolean with the default value `FALSE`, if the strings provided in `pattern` shall be matched exactly as it is or as a pattern.
`...`	additional arguments to the `grep` function like `ignore.case=TRUE` if the pattern should be searched case sensitive or not. The default value for `ignore.case` is `FALSE`.

Details

The function returns the line(s) where the searched pattern appears in the code or in the name column.

Value

If the pattern found then the function returns a data.frame with the 4 columns:

`pattern`	The pattern which was searched
`concept`	The name of the concepts in the data structure
`code`	The list of codes where the pattern was found, or the code of a name where the pattern appears
`name`	The name/description of the code where the pattern found, or the name of the code where the pattern appears

Otherwise returns the value `NULL`.

See Also

get_eurostat_dsd, create_filter_table, search_eurostat_toc

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
dsd_example<-get_eurostat_dsd("nama_10_gdp",verbose=TRUE)
search_eurostat_dsd("EU",dsd_example)
search_eurostat_dsd("EU",dsd_example,ignore.case=TRUE)
search_eurostat_dsd("EU27_2019",dsd_example,name=FALSE)
search_eurostat_dsd("EU27_2019",dsd_example,exact_match=TRUE)
options(timeout=60)
```

---

search_eurostat_toc *Search for pattern in the titles, units and short description of the TOC*

Description

Lists names of dataset from Eurostat with the particular pattern in the title, units or short description.

Usage

```
search_eurostat_toc(pattern, lang = "en", verbose = FALSE, ...)
```

Arguments

pattern	Character string to search for in the table of contents of Eurostat tables/datasets
lang	a character string either en, de or fr to define the language version for the table of contents. The default is en - English.
verbose	A boolean with default FALSE, so detailed messages (for debugging) will not printed. Can be set also with options(restatapi_verbose=TRUE)
...	other additional parameters to pass to the grepl function like ignore.case=TRUE if the pattern should be searched case sensitive or not. The default value for ignore.case is FALSE.

Details

Downloads the list of all tables and datasets available in the Eurostat database and returns all the details from the table of contents of the tables/datasets that contains particular pattern in the dataset title, unit or short description. E.g. all tables/datasets mentioning 'energy'.

Value

A table with the following columns:

title	The name of dataset/table in the language provided by the lang parameter
code	The codename of dataset/table which can be used by the get_eurostat function
type	The type of information: 'dataset' or 'table'
lastUpdate	The date when the data was last time updated for tables and datasets
lastModified	The date when the structure of the dataset/table was last time modified
dataStart	The start date of the data in the dataset/table
dataEnd	The end date of the data in the dataset/table
values	The number of values in the dataset/table
unit	The unit name for tables in the language provided by the lang parameter, if the type 'dataset' this colu
shortDescription	The short description of the values for tables in the language provided by the lang parameterif the type
metadata.html	The link to the metadata in html format
metadata.sdmx	The link to the metadata in SDMX format
downloadLink.tsv	The link to the whole dataset/table in tab separated values format in the bulk download facility

The value in the code column can be used as an id in the get_eurostat_data, get_eurostat_bulk, get_eurostat_raw and get_eurostat_dsd functions. If there is no hit for the search query, it returns NULL.

See Also

search_eurostat_dsd, get_eurostat_data, get_eurostat_toc

Examples

```
if (!(grepl("amzn|-aws|-azure ",Sys.info()['release']))) options(timeout=2)
head(search_eurostat_toc("energy",verbose=TRUE))
nrow(search_eurostat_toc("energy"))
head(search_eurostat_toc("energie",lang="de",ignore.case=TRUE))
nrow(search_eurostat_toc("energie",lang="de",ignore.case=TRUE))
options(timeout=60)
```

Index