Package 'eurostat'

February 9, 2022

Type Package

Title Tools for Eurostat Open Data

Version 3.7.10

Date 2022-02-08

Description Tools to download data from the Eurostat database <https://ec.europa.eu/eurostat> together with search and manipulation utilities.

License BSD_2_clause + file LICENSE

URL https://ropengov.github.io/eurostat/, https://github.com/rOpenGov/eurostat

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

Depends methods, R (>= 3.1.0)

Imports broom, classInt, countrycode, curl, dplyr, httr, jsonlite, lubridate, rappdirs, readr, RefManageR, regions, stringi, stringr, tibble, tidyr

Suggests RColorBrewer, knitr, rmarkdown, sf, sp, testthat (>= 3.0.0), remotes

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel false

Encoding UTF-8

LazyData true

MailingList rOpenGov <ropengov-forum@googlegroups.com>

NeedsCompilation no

Repository CRAN

RoxygenNote 7.1.2

X-schema.org-isPartOf http://ropengov.org/

X-schema.org-keywords ropengov

Author Leo Lahti [aut, cre] (<https://orcid.org/0000-0001-5537-637X>),
Janne Huovari [aut],
Markus Kainu [aut],
Przemyslaw Biecek [aut],
Daniel Antal [ctb],
Diego Hernangomez [ctb] (<https://orcid.org/0000-0001-8457-4658>),
Joona Lehtomaki [ctb],
Francois Briatte [ctb],
Reto Stauffer [ctb],
Paul Rougieux [ctb],
Anna Vasylytsya [ctb],
Oliver Reiter [ctb],
Pyry Kantanen [ctb] (<https://orcid.org/0000-0003-2853-2765>)

Maintainer Leo Lahti <leo.lahti@iki.fi>

Date/Publication 2022-02-09 11:30:02 UTC

R topics documented:

`eurostat-package` *R Tools for Eurostat open data*

Description

Brief summary of the eurostat package

Details

Package	eurostat
Type	Package
Version	3.7.10
Date	2014-2021
License	BSD_2_clause + file LICENSE
LazyLoad	yes

R Tools for Eurostat Open Data

regions functions

For working with sub-national statistics the basic functions of the regions package are imported https://regions.dataobservatory.eu/.

Author(s)

Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
```

```
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

`help("regions")`, https://regions.dataobservatory.eu/

Examples

```
library(eurostat)
```

`check_access_to_data` *Check access to ec.europe.eu*

Description

Check if R has access to resources at http://ec.europa.eu

Usage

```
check_access_to_data()
```

Value

a logical.

Author(s)

Markus Kainu markus.kainu@kapsi.fi

Examples

```
check_access_to_data()
```

clean_eurostat_cache *Clean Eurostat Cache*

Description

Delete all .rds files from the eurostat cache directory. See `get_eurostat()` for more on cache.

Usage

```
clean_eurostat_cache(cache_dir = NULL, config = FALSE)
```

Arguments

`cache_dir`	A path to cache directory. If `NULL` (default) tries to clean default temporary cache directory.
`config`	Logical `TRUE/FALSE`. Should the cached path be deleted?

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari, Markus Kainu and Diego Hernangómez

See Also

Other cache utilities: `set_eurostat_cache_dir()`

Examples

```
## Not run:
clean_eurostat_cache()

## End(Not run)
```

cut_to_classes *Cuts the Values Column into Classes and Polishes the Labels*

Description

Categorises a numeric vector into automatic or manually defined categories and polishes the labels ready for used in mapping with `ggplot2`.

Usage

```
cut_to_classes(
  x,
  n = 5,
  style = "equal",
  manual = FALSE,
  manual_breaks = NULL,
  decimals = 0,
  nodata_label = "No data"
)
```

Arguments

x	A numeric vector, eg. values variable in data returned by get_eurostat().
n	A numeric. number of classes/categories
style	chosen style: one of "fixed", "sd", "equal", "pretty", "quantile", "kmeans", "hclust", "bclust", "fisher", "jenks", "dpih" or "headtails"
manual	Logical. If manual breaks are being used
manual_breaks	Numeric vector with manual threshold values
decimals	Number of decimals to include with labels
nodata_label	String. Text label for NA category.

Value

a factor.

Author(s)

Markus Kainu markuskainu@gmail.com

See Also

classInt::classIntervals()

Other helpers: dic_order(), eurotime2date(), eurotime2num(), harmonize_country_code(), label_eurostat()

Examples

```
# lp <- get_eurostat("nama_aux_lp")
lp <- get_eurostat("nama_10_lp_ulc")
lp$class <- cut_to_classes(lp$values, n = 5, style = "equal", decimals = 1)
```

`dic_order` *Order of Variable Levels from Eurostat Dictionary.*

Description

Orders the factor levels.

Usage

```
dic_order(x, dic, type)
```

Arguments

`x`	a variable (code or labelled) to get order for.
`dic`	a name of the dictionary. Correspond a variable name in the data_frame from `get_eurostat()`. Can be also data_frame from `get_eurostat_dic()`.
`type`	a type of the x. Could be `code` or `label`.

Details

Some variables, like classifications, have logical or conventional ordering. Eurostat data tables are nor necessary ordered in this order. The function `dic_order()` get the ordering from Eurostat classifications dictionaries. The function `label_eurostat()` can also order factor levels of labels with argument `eu_order = TRUE`.

Value

A numeric vector of orders.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu

See Also

Other helpers: `cut_to_classes()`, `eurotime2date()`, `eurotime2num()`, `harmonize_country_code()`, `label_eurostat()`

`eurostat_geodata_60_2016` *Geospatial data of Europe from Gisco in 1:60 million scale from year 2016*

Description

Geospatial data of Europe from Gisco in 1:60 million scale from year 2016

Usage

```
eurostat_geodata_60_2016
```

Format

sf

- **id**: Country code in the Eurostat database.
- **CNTRY_CODE**: Country code.
- **NUTS_NAME**: NUTS name in local language.
- **LEVL_CODE**: NUTS code.
- **FID**: Country code.
- **NUTS_ID**: NUTS code.
- **geometry**: geospatial information.
- **geo**: NUTS code.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-uni

See Also

Other datasets: `tgs00026`

Other geospatial: `get_eurostat_geospatial()`

eurotime2date *Date Conversion from Eurostat Time Format*

Description

Date conversion from Eurostat time format. A function to convert Eurostat time values to objects of class `Date()` representing calendar dates.

Usage

```
eurotime2date(x, last = FALSE)
```

Arguments

x	a charter string with time information in Eurostat time format.
last	a logical. If FALSE (default) the date is the first date of the period (month, quarter or year). If TRUE the date is the last date of the period.

Value

an object of class `Date()`.

Author(s)

Janne Huovari janne.huovari@ptt.fi

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#      journal = {The R Journal},
#      volume = {9},
#      number = {1},
```

```
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

lubridate::ymd()

Other helpers: cut_to_classes(), dic_order(), eurotime2num(), harmonize_country_code(), label_eurostat()

Examples

```
na_q <- get_eurostat("namq_10_pc", time_format = "raw")
na_q$time <- eurotime2date(x = na_q$time)
unique(na_q$time)
```

---

eurotime2num *Conversion of Eurostat Time Format to Numeric*

Description

A conversion of a Eurostat time format to numeric.

Usage

```
eurotime2num(x)
```

Arguments

x	a charter string with time information in Eurostat time format.

Details

Bi-annual, quarterly and monthly data is presented as fraction of the year in beginning of the period. Conversion of daily data is not supported.

Value

see as.numeric().

Author(s)

Janne Huovari janne.huovari@ptt.fi

See Also

Other helpers: cut_to_classes(), dic_order(), eurotime2date(), harmonize_country_code(), label_eurostat()

Examples

```
na_q <- get_eurostat("namq_10_pc", time_format = "raw")
na_q$time <- eurotime2num(x = na_q$time)

unique(na_q$time)
```

---

eu_countries *Countries and Country Codes*

Description

Countries and country codes in EU, Euro area, EFTA and EU candidate countries.

Usage

```
eu_countries

ea_countries

efta_countries

eu_candidate_countries
```

Format

A data_frame:

- **code**: Country code in the Eurostat database.
- **name**: Country name in English.
- **label**: Country name in the Eurostat database.

An object of class `data.frame` with 19 rows and 3 columns.

An object of class `data.frame` with 4 rows and 3 columns.

An object of class `data.frame` with 5 rows and 3 columns.

Source

https://ec.europa.eu/eurostat/statistics-explained/index.php/Tutorial:Country_codes_and_protocol_order, https://ec.europa.eu/eurostat/statistics-explained/index.php/Glossary:Euro_area

get_bibentry *Create A Data Bibliography*

Description

Creates a bibliography from selected Eurostat data files, including last Eurostat update, URL access data, and optional keywords set by the user.

Usage

```
get_bibentry(code, keywords = NULL, format = "Biblatex")
```

Arguments

code	A Eurostat data code or a vector of Eurostat data codes as character or factor.
keywords	A list of keywords to be added to the entries. Defaults to NULL.
format	Default is 'Biblatex', alternatives are 'bibentry' or 'Bibtex' (not case sensitive.)

Value

a bibentry, Bibtex or Biblatex object.

Author(s)

Daniel Antal, Przemyslaw Biecek

Examples

```
my_bibliography <- get_bibentry(
  code = c("tran_hv_frtra", "t2020_rk310", "tec00001"),
  keywords = list(
    c("railways", "freight", "transport"),
    c("railways", "passengers", "modal split")
  ),
  format = "Biblatex"
)
my_bibliography
```

`get_eurostat` *Read Eurostat Data*

Description

Download data sets from Eurostat https://ec.europa.eu/eurostat/.

Usage

```
get_eurostat(
  id,
  time_format = "date",
  filters = "none",
  type = "code",
  select_time = NULL,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = FALSE,
  keepFlags = FALSE,
  ...
)
```

Arguments

`id`	A code name for the dataset of interest. See `search_eurostat()` or details for how to get code.
`time_format`	a string giving a type of the conversion of the time column from the eurostat format. A "date" (default) converts to a `Date()` with a first date of the period. A "date_last" converts to a `Date()` with a last date of the period. A "num" converts to a numeric and "raw" does not do conversion. See `eurotime2date()` and `eurotime2num()`.
`filters`	a "none" (default) to get a whole dataset or a named list of filters to get just part of the table. Names of list objects are Eurostat variable codes and values are vectors of observation codes. If `NULL` the whole dataset is returned via API. More on details. See more on filters and limitations per query via API from for `get_eurostat_json()`.
`type`	A type of variables, "code" (default) or "label".
`select_time`	a character symbol for a time frequency or NULL, which is used by default as most datasets have just one time frequency. For datasets with multiple time frequencies, select the desired time format with: Y = annual, S = semi-annual, Q = quarterly, M = monthly. For all frequencies in same data frame `time_format = "raw"` should be used.
`cache`	a logical whether to do caching. Default is `TRUE`. Affects only queries from the bulk download facility.

update_cache a logical whether to update cache. Can be set also with options(eurostat_update = TRUE)

cache_dir a path to a cache directory. The directory must exist. The NULL (default) uses and creates 'eurostat' directory in the temporary directory from tempdir(). The directory can also be set with set_eurostat_cache_dir().

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

stringsAsFactors if TRUE (the default) variables are converted to factors in original Eurostat order. If FALSE they are returned as a character.

keepFlags a logical whether the flags (e.g. "confidential", "provisional") 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/data/database/information. Also possible non-real zero "0n" is indicated in flags column. Flags are not available for eurostat API, so keepFlags can not be used with a filters.

... Arguments passed on to get_eurostat_json

lang A language used for metadata (en/fr/de).

Details

Data sets are downloaded from the Eurostat bulk download facility or from The Eurostat Web Services JSON API. If only the table id is given, the whole table is downloaded from the bulk download facility. If also filters are defined the JSON API is used.

The bulk download facility is the fastest method to download whole datasets. It is also often the only way as the JSON API has limitation of maximum 50 sub-indicators at time and whole datasets usually exceeds that. Also, it seems that multi frequency datasets can only be retrieved via bulk download facility and the select_time is not available for JSON API method.

If your connection is thru a proxy, you probably have to set proxy parameters to use JSON API, see get_eurostat_json().

By default datasets from the bulk download facility are cached as they are often rather large. Caching is not (currently) possible for datasets from JSON API. Cache files are stored in a temporary directory by default or in a named directory (See set_eurostat_cache_dir()). The cache can be emptied with clean_eurostat_cache().

The id, a code, for the dataset can be searched with the search_eurostat() or from the Eurostat database https://ec.europa.eu/eurostat/data/database. The Eurostat database gives codes in the Data Navigation Tree after every dataset in parenthesis.

Value

a tibble. One column for each dimension in the data, the time column for a time dimension and the values column for numerical values. Eurostat data does not include all missing values and a treatment of missing values depend on source. In bulk download facility missing values are dropped if all dimensions are missing on particular time. In JSON API missing values are dropped only if all dimensions are missing on all times. The data from bulk download facility can be completed for example with tidyr::complete().

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

`search_eurostat()`, `label_eurostat()`

Examples

```
k <- get_eurostat("nama_10_lp_ulc")
k <- get_eurostat("nama_10_lp_ulc", time_format = "num")
k <- get_eurostat("nama_10_lp_ulc", update_cache = TRUE)

k <- get_eurostat("nama_10_lp_ulc",
  cache_dir = file.path(tempdir(), "r_cache")
)
options(eurostat_update = TRUE)
k <- get_eurostat("nama_10_lp_ulc")
options(eurostat_update = FALSE)

set_eurostat_cache_dir(file.path(tempdir(), "r_cache2"))
```

```
k <- get_eurostat("nama_10_lp_ulc")
k <- get_eurostat("nama_10_lp_ulc", cache = FALSE)
k <- get_eurostat("avia_gonc", select_time = "Y", cache = FALSE)

## Not run:
dd <- get_eurostat("nama_10_gdp",
  filters = list(
    geo = "FI",
    na_item = "B1GQ",
    unit = "CLV_I10"
  )
)

## End(Not run)
```

`get_eurostat_dic` *Download Eurostat Dictionary*

Description

Download a Eurostat dictionary.

Usage

```
get_eurostat_dic(dictname, lang = "en")
```

Arguments

`dictname`	A character, dictionary for the variable to be downloaded.
`lang`	A character, language code. Options: "en" (default), "fr", "de".

Details

For given coded variable from Eurostat https://ec.europa.eu/eurostat/. The dictionaries link codes with human-readable labels. To translate codes to labels, use `label_eurostat()`.

Value

tibble with two columns: code names and full names.

Author(s)

Przemyslaw Biecek and Leo Lahti leo.lahti@iki.fi. Thanks to Wietse Dol for contributions.

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#      journal = {The R Journal},
#      volume = {9},
#      number = {1},
#      pages = {385--392},
#      year = {2017},
#      doi = {10.32614/RJ-2017-019},
#      url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

`label_eurostat()`, `get_eurostat()`, `search_eurostat()`.

Examples

```
get_eurostat_dic("crop_pro")

# Try another language
get_eurostat_dic("crop_pro", lang = "fr")
```

---

get_eurostat_geospatial

Download Geospatial Data from GISCO

Description

Downloads either a simple features (sf), SpatialPolygonDataFrame or a data_frame preprocessed using `broom::tidy()`.

Usage

```
get_eurostat_geospatial(
  output_class = "sf",
  resolution = "60",
  nuts_level = "all",
  year = "2016",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  crs = "4326",
  make_valid = FALSE
)
```

Arguments

output_class	A string. Class of object returned, either sf simple features, df (data_frame) or spdf (SpatialPolygonDataFrame)
resolution	Resolution of the geospatial data. One of • "60" (1:60million), • "20" (1:20million) • "10" (1:10million) • "03" (1:3million) or • "01" (1:1million).
nuts_level	Level of NUTS classification of the geospatial data. One of "0", "1", "2", "3" or "all" (mimics the original behaviour)
year	NUTS release year. One of "2003", "2006", "2010", "2013", "2016" or "2021"
cache	a logical whether to do caching. Default is TRUE. Affects only queries from the bulk download facility.
update_cache	a logical whether to update cache. Can be set also with options(eurostat_update = TRUE)
cache_dir	a path to a cache directory. The directory have to exist. The NULL (default) uses and creates 'eurostat' directory in the temporary directory from tempdir(). Directory can also be set with option eurostat_cache_dir.
crs	projection of the map: 4-digit EPSG code. One of: • "4326" - WGS84 • "3035" - ETRS89 / ETRS-LAEA • "3857" - Pseudo-Mercator
make_valid	logical; ensure that valid (multi-)polygon features are returned if output_class="sf", see Details. Current default FALSE, will be changed in the future.

Details

The data source URL is https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units. The source provides feature collections as line

strings (GeoJSON format), not as (multi-)polygons which, in some cases, yields invalid self-intersecting (multi-)polygon geometries (for some years/resolutions). This can cause problems, e.g., when using these geometries as input argument to `sf::st_interpolate_aw()`. `make_valid = TRUE` makes sure that only valid (multi-)polygons are returned, example included below.

Value

a sf, data_frame or SpatialPolygonDataFrame.

Author(s)

Markus Kainu markuskainu@gmail.com

See Also

Other geospatial: `eurostat_geodata_60_2016`

Examples

```
sf <- get_eurostat_geospatial(
  output_class = "sf",
  resolution = "60",
  nuts_level = "all"
)
df <- get_eurostat_geospatial(
  output_class = "df",
  resolution = "20",
  nuts_level = "0"
)


## Not run:
spdf <- get_eurostat_geospatial(
  output_class = "spdf",
  resolution = "10",
  nuts_level = "3"
)

## End(Not run)

## Not run:
# ----------------------------------------------------------------------
# Minimal example to demonstrate reason/effect of 'make_valid = TRUE'
# Spatial data set; rectangle spanning the entire globe with a constant value of 1L.
# Requires the R package sf.
library("sf")
d <- c(-180, -90, -180, 90, 180, 90, 180, -90, -180, -90)
poly <- st_polygon(list(matrix(d, ncol = 2, byrow = TRUE)))
data <- st_sf(data.frame(geom = st_sfc(poly), data = 1L),
  crs = st_crs(4326)
)
```

```
# Causing an error: Self-intersection of some points of the geometry
NUTS2_A <- get_eurostat_geospatial("sf", 60,
  nuts_level = 2, year = 2013,
  crs = 4326, make_valid = FALSE
)
res <- tryCatch(st_interpolate_aw(data, NUTS2_A, extensive = FALSE),
  error = function(e) e
)
print(res)

# Resolving the problem using
# make_valid = TRUE. 'extensive = FALSE' returns
# average over each area, thus resulting in a
# constant value of 1 for each geometry in NUTS2_B.
NUTS2_B <- get_eurostat_geospatial("sf", 60,
  nuts_level = 2, year = 2013,
  crs = 4326, make_valid = TRUE
)
res <- st_interpolate_aw(data, NUTS2_B, extensive = FALSE)
print(head(res))

## End(Not run)
```

`get_eurostat_json` *Get Data from Eurostat API in JSON*

Description

Retrieve data from Eurostat API in JSON format.

Usage

```
get_eurostat_json(
  id,
  filters = NULL,
  type = c("code", "label", "both"),
  lang = c("en", "fr", "de"),
  stringsAsFactors = FALSE,
  ...
)
```

Arguments

`id`	A code name for the dataset of interested. See the table of contents of eurostat datasets for more details.
`filters`	A named list of filters. Names of list objects are Eurostat variable codes and values are vectors of observation codes. If `NULL` (default) the whole dataset is returned. See details for more on filters and limitations per query.

`type` A type of variables, "code" (default), "label" or "both". The "both" will return a data_frame with named vectors, labels as values and codes as names.

`lang` A language used for metadata (en/fr/de).

`stringsAsFactors`
if `TRUE` (the default) variables are converted to factors in original Eurostat order. If `FALSE` they are returned as a character.

`...` Arguments passed on to `httr::GET`

`url` the url of the page to retrieve

`config` Additional configuration settings such as http authentication (`authenticate()`), additional headers (`add_headers()`), cookies (`set_cookies()`) etc. See `config()` for full details and list of helpers.

`handle` The handle to use with this request. If not supplied, will be retrieved and reused from the `handle_pool()` based on the scheme, hostname and port of the url. By default **httr** requests to the same scheme/host/port combo. This substantially reduces connection time, and ensures that cookies are maintained over multiple requests to the same host. See `handle_pool()` for more details.

Details

Data to retrieve from The Eurostat Web Services can be specified with filters. Normally, it is better to use JSON query through `get_eurostat()`, than to use `get_eurostat_json()` directly.

Queries are limited to 50 sub-indicators at a time. A time can be filtered with fixed "time" filter or with "sinceTimePeriod" and "lastTimePeriod" filters. A `sinceTimePeriod = 2000` returns observations from 2000 to a last available. A `lastTimePeriod = 10` returns a 10 last observations.

To use a proxy to connect, a `httr::use_proxy()` can be passed to `httr::GET()`. For example `get_eurostat_json(id,filters,config = httr::use_proxy(url,port,username,password))`.

Value

A dataset as a data_frame.

Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Markus Kainu

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
```

```
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

`httr::GET()`

Examples

```
# nama_gdp_c has been discontinued since 2/2018 and this example has ceased working.
## Not run:
tmp <- get_eurostat_json("cdh_e_fos")
yy <- get_eurostat_json(id = "nama_gdp_c", filters = list(
  geo = c("EU28", "FI"),
  unit = "EUR_HAB",
  indic_na = "B1GM"
))

## End(Not run)
```

---

get_eurostat_raw *Download Data from Eurostat Database*

Description

Download data from the eurostat database.

Usage

```
get_eurostat_raw(id)
```

Arguments

id	A code name for the dataset of interested. See the table of contents of eurostat datasets for more details.

Details

Data is downloaded from https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing and transformed into tabular format.

Value

A dataset in tibble format. First column contains comma separated codes of cases. Other columns usually corresponds to years and column names are years with preceding X. Data is in character format as it contains values together with eurostat flags for data.

Author(s)

Przemyslaw Biecek, Leo Lahti and Janne Huovari

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

get_eurostat().

Examples

```
eurostat:::get_eurostat_raw("educ_iste")
```

`get_eurostat_toc` *Download Table of Contents of Eurostat Data Sets*

Description

Download table of contents (TOC) of eurostat datasets.

Usage

```
get_eurostat_toc()
```

Details

The TOC is downloaded from https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?sort=1&file=table_of_contents_en.txt. The values in column 'code' should be used to download a selected dataset.

Value

A tibble with eight columns:

- title: The name of dataset of theme.
- code: The codename of dataset of theme, will be used by the `get_eurostat()` and `get_eurostat_raw()` functions.
- type: Is it a dataset, folder or table.
- last.update.of.data, last.table.structure.change, data.start, data.end: Dates.

Author(s)

Przemyslaw Biecek and Leo Lahti ropengov-forum@googlegroups.com

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
```

```
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

get_eurostat(), search_eurostat().

Examples

```
tmp <- get_eurostat_toc()
head(tmp)
```

---

harmonize_country_code

Harmonize Country Code

Description

The European Commission and the Eurostat generally uses ISO 3166-1 alpha-2 codes with two exceptions: EL (not GR) is used to represent Greece, and UK (not GB) is used to represent the United Kingdom. This function turns country codes into to ISO 3166-1 alpha-2.

Usage

```
harmonize_country_code(x)
```

Arguments

x	A character or a factor vector of eurostat countycodes.

Value

a vector.

Author(s)

Janne Huovari janne.huovari@ptt.fi

See Also

Other helpers: `cut_to_classes()`, `dic_order()`, `eurotime2date()`, `eurotime2num()`, `label_eurostat()`

Examples

```
lp <- get_eurostat("nama_10_lp_ulc")
lp$geo <- harmonize_country_code(lp$geo)
```

`label_eurostat` *Get Eurostat Codes*

Description

Get definitions for Eurostat codes from Eurostat dictionaries.

Usage

```
label_eurostat(
  x,
  dic = NULL,
  code = NULL,
  eu_order = FALSE,
  lang = "en",
  countrycode = NULL,
  countrycode_nomatch = NULL,
  custom_dic = NULL,
  fix_duplicated = FALSE
)

label_eurostat_vars(x, lang = "en")

label_eurostat_tables(x, lang = "en")
```

Arguments

`x`	A character or a factor vector or a data_frame.
`dic`	A string (vector) naming eurostat dictionary or dictionaries. If `NULL` (default) dictionary names taken from column names of the data_frame.

code	For data_frames names of the column for which also code columns should be retained. The suffix "_code" is added to code column names.
eu_order	Logical. Should Eurostat ordering used for label levels. Affects only factors.
lang	A character, code for language. Available are "en" (default), "fr" and "de".
countrycode	A NULL or a name of the coding scheme for the countrycode::countrycode() to label "geo" variable with countrycode-package. It can be used to convert to short and long country names in many different languages. If NULL (default) eurostat dictionary is used instead.
countrycode_nomatch	What to do when using the countrycode to label a "geo" and countrycode fails to find a match, for example other than country codes like EU28. The original code is used with a NULL (default), eurostat dictionary label is used with "eurostat", and NA is used with NA.
custom_dic	a named vector or named list of named vectors to give an own dictionary for (part of) codes. Names of the vector should be codes and values labels. List can be used to specify dictionaries and then list names should be dictionary codes.
fix_duplicated	A logical. If TRUE, the code is added to the duplicated label values. If FALSE (default) error is given if labeling produce duplicates.

Details

A character or a factor vector of codes returns a corresponding vector of definitions. label_eurostat() labels also data_frames from get_eurostat(). For vectors a dictionary name have to be supplied. For data_frames dictionary names are taken from column names. "time" and "values" columns are returned as they were, so you can supply data_frame from get_eurostat() and get data_frame with definitions instead of codes.

Some Eurostat dictionaries includes duplicated labels. By default duplicated labels cause an error, but they can be fixed automatically with fix_duplicated = TRUE.

Value

a vector or a data_frame.

Functions

- label_eurostat_vars: Get definitions for variable (column) names. For objects other than characters or factors definitions are get for names.
- label_eurostat_tables: Get definitions for table names

Author(s)

Janne Huovari janne.huovari@ptt.fi

See Also

countrycode::countrycode()

Other helpers: cut_to_classes(), dic_order(), eurotime2date(), eurotime2num(), harmonize_country_code()

Examples

```
## Not run:
lp <- get_eurostat("nama_10_lp_ulc")
lpl <- label_eurostat(lp)
str(lpl)
lpl_order <- label_eurostat(lp, eu_order = TRUE)
lpl_code <- label_eurostat(lp, code = "unit")
label_eurostat_vars(names(lp))
label_eurostat_tables("nama_10_lp_ulc")
label_eurostat(c("FI", "DE", "EU28"), dic = "geo")
label_eurostat(c("FI", "DE", "EU28"), dic = "geo", custom_dic = c(DE = "Germany"))
label_eurostat(c("FI", "DE", "EU28"),
  dic = "geo", countrycode = "country.name",
  custom_dic = c(EU28 = "EU")
)
label_eurostat(c("FI", "DE", "EU28"), dic = "geo", countrycode = "country.name")
# In Finnish
label_eurostat(c("FI", "DE", "EU28"), dic = "geo", countrycode = "cldr.short.fi")

## End(Not run)
```

---

search_eurostat *Grep Datasets Titles from Eurostat*

Description

Lists names of dataset from eurostat with the particular pattern in the description.

Usage

```
search_eurostat(pattern, type = "dataset", fixed = TRUE)

grepEurostatTOC(pattern, type = "dataset")
```

Arguments

pattern	Character, datasets, folder or tables with this pattern in the description will be returned (depending on the 'type' argument)
type	Grep the Eurostat table of contents either for 'dataset' (default), 'folder', 'table' or "all" (for all types).
fixed	logical. If TRUE, pattern is a string to be matched as is. Change to FALSE if more complex regex matching is needed.

Details

Downloads list of all datasets available on eurostat and return list of names of datasets that contains particular pattern in the dataset description. E.g. all datasets related to education of teaching.

Value

A tibble with eight columns

- **title**: The name of dataset of theme
  - **code**: The codename of dataset of theme, will be used by the `get_eurostat()` and `get_eurostat_raw()` functions.
  - **type**: Is it a dataset, folder or table.
  - **last.update.of.data, last.table.structure.change, data.start, data.end**: Dates.

Functions

- `grepEurostatTOC`: Old deprecated version

Author(s)

Przemyslaw Biecek and Leo Lahti ropengov-forum@googlegroups.com

References

See `citation("eurostat")`:

```
#
# Kindly cite the eurostat R package as follows:
#
#   (C) Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek.
#   Retrieval and analysis of Eurostat open data with the eurostat
#   package. R Journal 9(1):385-392, 2017. doi: 10.32614/RJ-2017-019
#   Package URL: http://ropengov.github.io/eurostat Article URL:
#   https://journal.r-project.org/archive/2017/RJ-2017-019/index.html
#
# A BibTeX entry for LaTeX users is
#
#   @Article{,
#    title = {Retrieval and Analysis of Eurostat Open Data with the eurostat Package},
#    author = {Leo Lahti and Janne Huovari and Markus Kainu and Przemyslaw Biecek},
#     journal = {The R Journal},
#     volume = {9},
#     number = {1},
#     pages = {385--392},
#     year = {2017},
#     doi = {10.32614/RJ-2017-019},
#     url = {https://doi.org/10.32614/RJ-2017-019},
#   }
```

See Also

`get_eurostat()`, `get_eurostat_toc()`

Examples

```
tmp <- search_eurostat("education")
head(tmp)
# Use "fixed = TRUE" when pattern has characters that would need escaping.
# Here, parentheses would normally need to be escaped in regex
tmp <- search_eurostat("Live births (total) by NUTS 3 region", fixed = TRUE)
```

---

set_eurostat_cache_dir

Set Eurostat Cache

Description

This function will store your cache_dir path on your local machine and would load it for future sessions. Type Sys.getenv("EUROSTAT_CACHE_DIR") to find your cached path.

Alternatively, you can store the cache_dir manually with the following options:

- Run Sys.setenv(EUROSTAT_CACHE_DIR = "cache_dir"). You would need to run this command on each session (Similar to install = FALSE).
- Set options(eurostat_cache_dir = "cache_dir"). Similar to the previous option. This is provided for backwards compatibility purposes.
- Write this line on your .Renviron file: EUROSTAT_CACHE_DIR = "value_for_cache_dir" (same behavior than install = TRUE). This would store your cache_dir permanently.

Usage

```
set_eurostat_cache_dir(
  cache_dir,
  overwrite = FALSE,
  install = FALSE,
  verbose = TRUE
)
```

Arguments

cache_dir	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See base::tempdir()).
overwrite	If this is set to TRUE, it will overwrite an existing EUROSTAT_CACHE_DIR that you already have in local machine.
install	if TRUE, will install the key in your local machine for use in future sessions. Defaults to FALSE. If cache_dir is FALSE this parameter is set to FALSE automatically.
verbose	Logical, displays information. Useful for debugging, default is FALSE.

Value

An (invisible) character with the path to your `cache_dir`.

Author(s)

Diego Hernangómez

See Also

`rappdirs::user_config_dir()`

Other cache utilities: `clean_eurostat_cache()`

Examples

```
# Don't run this! It would modify your current state
## Not run:
set_eurostat_cache_dir(verbose = TRUE)

## End(Not run)

Sys.getenv("EUROSTAT_CACHE_DIR")
```

---

tgs00026 *Auxiliary Data*

Description

Auxiliary Data Sets

Usage

```
tgs00026
```

Format

data_frame

Details

Retrieved with: `tgs00026 <-get_eurostat("tgs00026",time_format = "raw")`

See Also

Other datasets: `eurostat_geodata_60_2016`

Index