# Package 'eurostat'

May 8, 2026

**Type** Package

**Title** Tools for Eurostat Open Data

**Version** 4.0.0

**Date** 2023-12-19

**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** R (>= 3.6.0)

**Imports** classInt, countrycode, curl, digest, dplyr, httr2 (>= 0.2.3), ISOweek, jsonlite, lubridate, rappdirs, readr, RefManageR, regions, rlang, stringi, stringr, tibble, tidyr (>= 1.0.0), xml2, data.table (>= 1.14.8)

**Suggests** giscoR, knitr, rmarkdown, sf, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/Needs/website** ggplot2, tmap, styler, sessioninfo, ropengov/rogtemplate, ragg

**Config/testthat/edition** 3

**Config/testthat/parallel** false

**Encoding** UTF-8

**LazyData** true

**MailingList** rOpenGov <ropengov-forum@googlegroups.com>

**NeedsCompilation** no

**Repository** CRAN

**RoxygenNote** 7.2.3

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

**X-schema.org-keywords** ropengov

**Author** Leo Lahti [aut, cre] (ORCID: <https://orcid.org/0000-0001-5537-637X>),
Janne Huovari [aut],
Markus Kainu [aut],
Przemyslaw Biecek [aut],
Daniel Antal [ctb],
Diego Hernangomez [ctb] (ORCID:
<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] (ORCID: <https://orcid.org/0000-0003-2853-2765>),
Enrico Spinielli [ctb] (ORCID: <https://orcid.org/0000-0001-8584-9131>)

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

**Date/Publication** 2023-12-19 20:30:02 UTC

# Contents

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

## Description

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

## Details

| | |
|---|---|
| **Package** | eurostat |
| **Type** | Package |
| **Version** | 4.0.0 |
| **Date** | 2014-2023 |
| **License** | BSD_2_clause + file LICENSE |
| **LazyLoad** | yes |

## Eurostat

Eurostat website: https://ec.europa.eu/eurostat Eurostat database: https://ec.europa.eu/eurostat/web/main/data/database

Information about the data update schedule from Eurostat: "Eurostat datasets are updated twice a day at 11:00 and 23:00 CET, if newer data is available or for structural changes, for example for the dimensions in the dataset.

The Eurostat database always contains the latest version of the datasets, meaning that there is no versioning or documentation of past versions of the data."

## Data source: Eurostat SDMX 2.1 Dissemination API

Data is downloaded from Eurostat SDMX 2.1 API endpoint as compressed TSV files that are transformed into tabular format. See Eurostat documentation for more information: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+query

The new dissemination API replaces the old bulk download facility that was used by Eurostat before October 2023 and by the eurostat R package versions before 4.0.0. See Eurostat documentation about the transition from Bulk Download to API for more information about the differences between the old bulk download facility and the data provided by the new API connection: https://wikis.ec.europa.eu/display/EUROSTATHELP/Transition+-+from+Eurostat+Bulk+Download+to+API

See especially the document Migrating_to_API_TSV.pdf that describes the changes in TSV file format in new applications.

For more information about SDMX 2.1, see SDMX standards: Section 7: Guidelines for the use of web services, Version 2.1: https://sdmx.org/wp-content/uploads/SDMX_2-1_SECTION_7_WebServicesGuidelines.pdf

### Disclaimer: Availability of filtering functionalities

Currently it only possible to download filtered data through API Statistics (JSON API) when using `eurostat` package, although technically filtering datasets downloaded through the SDMX Dissemination API is also supported by Eurostat. We may support this feature in the future. In the meantime, if you are interested in filtering Dissemination API data queries manually, please consult the following Eurostat documentation: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+filtering

### Data source: Eurostat API Statistics (JSON API)

Data is downloaded from Eurostat API Statistics. See Eurostat documentation for more information about data queries in API Statistics https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+data+query

This replaces the old JSON Web Services that was used by Eurostat before February 2023 and by the eurostat R package versions before 3.7.13. See Eurostat documentation about the migration from JSON web service to API Statistics for more information about the differences between the old and the new service: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+migrating+from+JSON+web+service+to+API+Statistics

For easily viewing which filtering options are available - in addition to the default ones, time and language - Eurostat Web services Query builder tool may be useful: https://ec.europa.eu/eurostat/web/query-builder

### Filtering datasets

When using Eurostat API Statistics (JSON API), datasets can be filtered before they are downloaded and saved in local memory. The general format for filter parameters is `<DIMENSION_CODE>=<VALUE>`.

Filter parameters are optional but the used dimension codes must be present in the data product that is being queried. Dimension codes can vary between different data products so it may be useful to examine new datasets in Eurostat data browser beforehand. However, most if not all Eurostat datasets concern European countries and contain information that was gathered at some point in time, so `geo` and `time` dimension codes can usually be used.
`<DIMENSION_CODE>` and `<VALUE>` are case-insensitive and they can be written in lowercase or uppercase in the query.

Parameters are passed onto the `eurostat` package functions `get_eurostat()` and `get_eurostat_json()` as a list item. If an individual item contains multiple items, as it often can be in the case of `geo` parameters and other optional items, they must be in the form of a vector: `c("FI", "SE")`. For examples on how to use these parameters, see function examples below.

#### Time parameters:

`time` and `time_period` address the same `TIME_PERIOD` dimension in the dataset and can be used interchangeably. In the Eurostat documentation it is stated that "Using more than one Time parameter in the same query is not accepted", but practice has shown that actually Eurostat API allows multiple `time` parameters in the same query. This makes it possible to use R colon operator when writing queries, so `time = c(2015:2018)` translates to `&time=2015&time=2016&time=2017&time=2018`.

The only exception to this is when the queried dataset contains e.g. quarterly data and `TIME_PERIOD` is saved as `2015-Q1`, `2015-Q2` etc. Then it is possible to use `time=2015-Q1&time=2015-Q2` style

in the query URL, but this makes it unfeasible to use the colon operator and requires a lot of manual typing.

Because of this, it is useful to know about other time parameters as well:

- `untilTimePeriod`: return dataset items from the oldest record up until the set time, for example "all data until 2000": `untilTimePeriod = 2000`
- `sinceTimePeriod`: return dataset items starting from set time, for example "all datastarting from 2008": `sinceTimePeriod = 2008`
- `lastTimePeriod`: starting from the most recent time period, how many preceding time periods should be returned? For example 10 most recent observations: `lastTimePeriod = 10`

Using both `untilTimePeriod` and `sinceTimePeriod` parameters in the same query is allowed, making the usage of the R colon operator unnecessary. In the case of quarterly data, using `untilTimePeriod` and `sinceTimePeriod` parameters also works, as opposed to the colon operator, so it is generally safer to use them as well.

**Other dimensions:**

In `get_eurostat_json()` examples `nama_10_gdp` dataset is filtered with two additional filter parameters:

- `na_item = "B1GQ"`
- `unit = "CLV_I10"`

Filters like these are most likely unique to the `nama_10_gdp` dataset (or other datasets within the same domain) and should not be used with others dataset without user discretion. By using `label_eurostat()` we know that `"B1GQ"` stands for "Gross domestic product at market prices" and `"CLV_I10"` means "Chain linked volumes, index 2010=100".

Different dimension codes can be translated to a natural language by using the `get_eurostat_dic()` function, which returns labels for individual dimension items such as `na_item` and `unit`, as opposed to `label_eurostat()` which does it for whole datasets. For example, the parameter `na_item` stands for "National accounts indicator (ESA 2010)" and `unit` stands for "Unit of measure".

**Language:**

All datasets have metadata available in English, French and German. If no parameter is given, the labels are returned in English.

Example:

- `lang = "fr"`

**More information:**

For more information about data filtering see Eurostat documentation on API Statistics: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+data+query#APIStatisticsdataquery-Thepara

## Data source: Eurostat Table of Contents

The Eurostat Table of Contents (TOC) is downloaded from https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=en (default) or from French or German language variants: https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=fr https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=de

See Eurostat documentation on TOC items: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+-+Detailed+guidelines+-+Catalogue+API+-+TOC

**Data source: GISCO - General Copyright**

"Eurostat's general copyright notice and licence policy is applicable and can be consulted here: https://ec.europa.eu/eurostat/about-us/policies/copyright

Please also be aware of the European Commission's general conditions: https://commission.europa.eu/legal-notice_en

Moreover, there are specific provisions applicable to some of the following datasets available for downloading. The download and usage of these data is subject to their acceptance:

- Administrative Units / Statistical Units
- Population distribution / Demography
- Transport Networks
- Land Cover
- Elevation (DEM)"

Of the abovementioned datasets, Administrative Units / Statistical Units is applicable if the user wants to draw maps with borders provided by GISCO / EuroGeographics.

**Data source: GISCO - Administrative Units / Statistical Units**

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the GISCO website: GISCO: Geographical information and maps - Administrative units/statistical units

"In addition to the general copyright and licence policy applicable to the whole Eurostat website, the following specific provisions apply to the datasets you are downloading. The download and usage of these data is subject to the acceptance of the following clauses:

1. The Commission agrees to grant the non-exclusive and not transferable right to use and process the Eurostat/GISCO geographical data downloaded from this page (the "data").
2. The permission to use the data is granted on condition that:
   (a) the data will not be used for commercial purposes;
   (b) the source will be acknowledged. A copyright notice, as specified below, will have to be visible on any printed or electronic publication using the data downloaded from this page."

**Copyright notice:**

When data downloaded from this page is used in any printed or electronic publication, in addition to any other provisions applicable to the whole Eurostat website, data source will have to be acknowledged in the legend of the map and in the introductory page of the publication with the following copyright notice:

EN: © EuroGeographics for the administrative boundaries

FR: © EuroGeographics pour les limites administratives

DE: © EuroGeographics bezüglich der Verwaltungsgrenzen

For publications in languages other than English, French or German, the translation of the copyright notice in the language of the publication shall be used.

If you intend to use the data commercially, please contact EuroGeographics for information regarding their licence agreements."

### Eurostat: Copyright notice and free re-use of data

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: https://ec.europa.eu/eurostat/about-us/policies/copyright

"(c) European Union, 1995 - today

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat;
- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information."

For exceptions to the abovementioned principles see Eurostat website

### Citing Eurostat data

For citing datasets, use `get_bibentry()` to build a bibliography that is suitable for your reference manager of choice.

When using Eurostat data in other contexts than academic publications that in-text citations or footnotes/endnotes, the following guidelines may be helpful:

- The origin of the data should always be mentioned as "Source: Eurostat".
- The online dataset codes(s) should also be provided in order to ensure transparency and facilitate access to the Eurostat data and related methodological information. For example: "Source: Eurostat (online data code: namq_10_gdp)"
- Online publications (e.g. web pages, PDF) should include a clickable link to the dataset using the bookmark functionality available in the Eurostat data browser.

It should be avoided to associate different entities (e.g. Eurostat, National Statistical Offices, other data providers) to the same dataset or indicator without specifying the role of each of them in the treatment of data.

See also section "Eurostat: Copyright notice and free re-use of data" in `get_eurostat()` documentation.

### Strategies for handling large datasets more efficiently

Most Eurostat datasets are relatively manageable, at least on a machine with 16 GB of RAM. The largest dataset in Eurostat database, at the time of writing this, had 148362539 (148 million) values, which results in an object with 148 million rows in tidy data (long) format. The test machine with 16 GB of RAM was able to handle the second largest dataset in the database with 91 million values (rows).

There are still some methods to make data fetching functions perform faster:

- turn caching off: `get_eurostat(cache = FALSE)`
- turn cache compression off (may result in rather large cache files!): `get_eurostat(compress_file = FALSE)`

- if you want faster caching with manageable file sizes, use stringsAsFactors: `get_eurostat(cache = TRUE, compress_file = TRUE, stringsAsFactors = TRUE)`
- Use faster data.table functions: `get_eurostat(use.data.table = TRUE)`
- Keep column processing to a minimum: `get_eurostat(time_format = "raw", type = "code")` etc.
- Read `get_eurostat()` function documentation carefully so you understand what different arguments do
- Filter the dataset so that you fetch only the parts you need!

### 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:

  Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
  analysis of Eurostat open data with the eurostat package. The R
  Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019

  Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
  and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
  [Computer software]. R package version 4.0.0.
  https://github.com/rOpenGov/eurostat

To see these entries in BibTeX format, use 'print(<citation>,
bibtex=TRUE)', 'toBibtex(.)', or set
'options(citation.bibtex.max=999)'.
```

When citing data downloaded from Eurostat, see section "Citing Eurostat data" in `get_eurostat()` documentation.

### 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", "headtails", "maximum", or "box" |
| 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-defunct *Defunct functions in eurostat*

---

## Description

This list of defunct functions is maintained to document changes to eurostat functions in a transparent manner.

## Usage

```
grepEurostatTOC(...)
```

## Arguments

| | |
|---|---|
| ... | Generic representation of old arguments |

## Details

The following functions are defunct:

- `grepEurostatTOC`: Use `search_eurostat` instead

---

# 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

## Format

sf object

### Details

The dataset contains 2016 observations (rows) and 12 variables (columns).

The object contains the following columns:

- **id**: JSON id code, the same as **NUTS_ID**. See **NUTS_ID** below for further clarification.
- **LEVL_CODE**: NUTS level code: 0 (national level), 1 (major socio-economic regions), 2 (basic regions for the application of regional policies) or 3 (small regions).
- **NUTS_ID**: NUTS ID code, consisting of country code and numbers (1 for NUTS 1, 2 for NUTS 2 and 3 for NUTS 3)
- **CNTR_CODE**: Country code: two-letter ISO code (ISO 3166 alpha-2), except in the case of Greece (EL).
- **NAME_LATN**: NUTS name in local language, transliterated to Latin script
- **NUTS_NAME**: NUTS name in local language, in local script.
- **MOUNT_TYPE**: Mountain typology for NUTS 3 regions.
  - 1: "where more than 50 % of the surface is covered by topographic mountain areas"
  - 2: "in which more than 50 % of the regional population lives in topographic mountain areas"
  - 3: "where more than 50 % of the surface is covered by topographic mountain areas and where more than 50 % of the regional population lives in these mountain areas"
  - 4: non-mountain region / other region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 and non-EU countries)
- **URBN_TYPE**: Urban-rural typology for NUTS 3 regions.
  - 1: predominantly urban region
  - 2: intermediate region
  - 3: predominantly rural region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 regions)
- **COAST_TYPE**: Coastal typology for NUTS 3 regions.
  - 1: coastal (on coast)
  - 2: coastal (>= 50% of population living within 50km of the coastline)
  - 3: non-coastal region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 regions)
- **FID**: Same as NUTS_ID.
- **geo**: Same as NUTS_ID, added for for easier joins with dplyr. However, it is recommended to use other identical fields for this purpose.
- **geometry**: geospatial information.

Dataset updated: 2023-06-29. For a more recent version, please use `giscoR::gisco_get_nuts()` function.

## Source

Data source: Eurostat via `giscoR::gisco_get_nuts()`.

© EuroGeographics for the administrative boundaries

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

## References

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: GISCO: Geographical information and maps - Administrative units/statistical units

"In addition to the general copyright and licence policy applicable to the whole Eurostat website, the following specific provisions apply to the datasets you are downloading. The download and usage of these data is subject to the acceptance of the following clauses:

1. The Commission agrees to grant the non-exclusive and not transferable right to use and process the Eurostat/GISCO geographical data downloaded from this page (the "data").
2. The permission to use the data is granted on condition that:
   (a) the data will not be used for commercial purposes;
   (b) the source will be acknowledged. A copyright notice, as specified below, will have to be visible on any printed or electronic publication using the data downloaded from this page.

**Copyright notice:**

When data downloaded from this page is used in any printed or electronic publication, in addition to any other provisions applicable to the whole Eurostat website, data source will have to be acknowledged in the legend of the map and in the introductory page of the publication with the following copyright notice:

EN: © EuroGeographics for the administrative boundaries

FR: © EuroGeographics pour les limites administratives

DE: © EuroGeographics bezüglich der Verwaltungsgrenzen

For publications in languages other than English, French or German, the translation of the copyright notice in the language of the publication shall be used.

If you intend to use the data commercially, please contact EuroGeographics for information regarding their licence agreements."

## See Also

`giscoR::gisco_get_nuts()` and Eurostat. (2019). Methodological manual on territorial typologies – 2018 edition. Manuals and guidelines.

Other datasets: `eu_countries`, `tgs00026`

Other geospatial: `get_eurostat_geospatial()`

## Examples

```
eurostat_geodata_60_2016 <- eurostat::eurostat_geodata_60_2016

# Manipulate and plot
if (require(sf)) {
  library(sf)
  # Filter NUTS3 from select countries like in a regular data frame
  example_nuts <- subset(eurostat_geodata_60_2016, LEVL_CODE == 3 &
    CNTR_CODE %in% c("DK", "DE", "PL"))

  plot(example_nuts["CNTR_CODE"])
}
```

---

# eurotime2date *Date Conversion from New 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. |

## Details

Available patterns are YYYY (year), YYYY-SN (semester), YYYY-QN (quarter), YYYY-MM (month), YYYY-WNN (week) and YYYY-MM-DD (day).

## 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:
#
#   Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
#   analysis of Eurostat open data with the eurostat package. The R
#   Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019
#
#   Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
#   and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
#   [Computer software]. R package version 4.0.0.
#   https://github.com/rOpenGov/eurostat
#
# To see these entries in BibTeX format, use 'print(<citation>,
# bibtex=TRUE)', 'toBibtex(.)', or set
# 'options(citation.bibtex.max=999)'.
```

## 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$TIME_PERIOD <- eurotime2date(x = na_q$TIME_PERIOD)
unique(na_q$TIME_PERIOD)


## Not run:
# Test for weekly data
get_eurostat(
  id = "lfsi_abs_w",
  select_time = c("W"),
  time_format = "date"
  )

## End(Not run)
```

---

`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 (semester), quarterly, monthly and weekly data can be presented as a 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, Pyry Kantanen

## 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$TIME_PERIOD <- eurotime2num(x = na_q$TIME_PERIOD)

unique(na_q$TIME_PERIOD)
```

---

# 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 `tbl_df` (inherits from `tbl`, `data.frame`) with 19 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 4 rows and 3 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7 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

## See Also

Other datasets: eurostat_geodata_60_2016, tgs00026

---

`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", lang = "en")
```

## 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) |
| lang | 2-letter language code, default is "en" (English), other options are "fr" (French) and "de" (German). Used for labeling datasets. |

## Value

a bibentry, Bibtex or Biblatex object.

## Citing Eurostat data

For citing datasets, use `get_bibentry()` to build a bibliography that is suitable for your reference manager of choice.

When using Eurostat data in other contexts than academic publications that in-text citations or footnotes/endnotes, the following guidelines may be helpful:

- The origin of the data should always be mentioned as "Source: Eurostat".
- The online dataset codes(s) should also be provided in order to ensure transparency and facilitate access to the Eurostat data and related methodological information. For example: "Source: Eurostat (online data code: namq_10_gdp)"
- Online publications (e.g. web pages, PDF) should include a clickable link to the dataset using the bookmark functionality available in the Eurostat data browser.

It should be avoided to associate different entities (e.g. Eurostat, National Statistical Offices, other data providers) to the same dataset or indicator without specifying the role of each of them in the treatment of data.

See also section "Eurostat: Copyright notice and free re-use of data" in `get_eurostat()` documentation.

## Author(s)

Daniel Antal, Przemyslaw Biecek

## See Also

utils::bibentry RefManageR::toBiblatex

## Examples

```
## Not run:
  my_bibliography <- get_bibentry(
    code = c("tran_hv_frtra", "tec00001"),
    keywords = list(
      c("transport", "freight", "multimodal data", "GDP"),
      c("economy and finance", "annual", "national accounts", "GDP")
```

```
  ),
  format = "Biblatex"
)
my_bibliography

## End(Not run)
```

---

`get_eurostat` *Get Eurostat Data*

---

## Description

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

## Usage

```
get_eurostat(
  id,
  time_format = "date",
  filters = NULL,
  type = "code",
  select_time = NULL,
  lang = "en",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  compress_file = TRUE,
  stringsAsFactors = FALSE,
  keepFlags = FALSE,
  use.data.table = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| `id` | A unique identifier / code for the dataset of interest. If code is not known `search_eurostat()` function can be used to search Eurostat table of contents. |
| `time_format` | a string giving a type of the conversion of the time column from the eurostat format. The default argument "`date`" converts to a `Date()` class with the date being the first day of the period. A "`date_last`" argument converts the dataset date to a `Date()` class object with the difference that the exact date is the last date of the period. Period can be year, semester (half year), quarter, month, or week (See `eurotime2date()` for more information). Argument "`num`" converts the date into a numeric (integer) meaning that the first day of the year 2000 is close to 2000.01 and the last day of the year is close to 2000.99 (see `eurotime2num()` for more information). Using the argument "`raw`" preserves the dates as they were in the original Eurostat data. |

`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 information on filters and limitations per query.

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

`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 one or more of the desired frequencies with: "Y" (or "A") = annual, "S" = semi-annual / semester, "Q" = quarterly, "M" = monthly, "W" = weekly. For all frequencies in same data frame `time_format = "raw"` should be used.

`lang` 2-letter language code, default is "`en`" (English), other options are "`fr`" (French) and "`de`" (German). Used for labeling datasets.

`cache` a logical whether to do caching. Default is `TRUE`.

`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. `NULL` (default) uses and creates 'eurostat' directory in the temporary directory defined by base R `tempdir()` function. The user can set the cache directory to an existing directory by using this argument. The cache directory can also be set with `set_eurostat_cache_dir()` function.

`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 the original Eurostat order. If `FALSE` they are returned as strings.

`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`.

`use.data.table` Use faster data.table functions? Default is FALSE. On Windows requires that RTools is installed.

`...` Arguments passed on to `get_eurostat_json`

`proxy` Use proxy, TRUE or FALSE (default).

## Details

Datasets are downloaded from the Eurostat SDMX 2.1 API in TSV format or from The Eurostat API Statistics JSON API. If only the table `id` is given, the whole table is downloaded from the SDMX API. If any `filters` are given JSON API is used instead.

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 through a proxy, you may have to set proxy parameters to use JSON API, see `get_eurostat_json()`.

By default datasets are cached to reduce load on Eurostat services and because some datasets can be quite large. 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()`.

## Eurostat: Copyright notice and free re-use of data

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: `https://ec.europa.eu/eurostat/about-us/policies/copyright`

"(c) European Union, 1995 - today

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat;
- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information."

For exceptions to the abovementioned principles see Eurostat website

## Filtering datasets

When using Eurostat API Statistics (JSON API), datasets can be filtered before they are downloaded and saved in local memory. The general format for filter parameters is `<DIMENSION_CODE>=<VALUE>`.

Filter parameters are optional but the used dimension codes must be present in the data product that is being queried. Dimension codes can vary between different data products so it may be useful to examine new datasets in Eurostat data browser beforehand. However, most if not all Eurostat datasets concern European countries and contain information that was gathered at some point in time, so `geo` and `time` dimension codes can usually be used.
`<DIMENSION_CODE>` and `<VALUE>` are case-insensitive and they can be written in lowercase or uppercase in the query.

Parameters are passed onto the eurostat package functions get_eurostat() and get_eurostat_json() as a list item. If an individual item contains multiple items, as it often can be in the case of geo parameters and other optional items, they must be in the form of a vector: c("FI", "SE"). For examples on how to use these parameters, see function examples below.

**Time parameters:**

time and time_period address the same TIME_PERIOD dimension in the dataset and can be used interchangeably. In the Eurostat documentation it is stated that "Using more than one Time parameter in the same query is not accepted", but practice has shown that actually Eurostat API allows multiple time parameters in the same query. This makes it possible to use R colon operator when writing queries, so time = c(2015:2018) translates to &time=2015&time=2016&time=2017&time=2018.

The only exception to this is when the queried dataset contains e.g. quarterly data and TIME_PERIOD is saved as 2015-Q1, 2015-Q2 etc. Then it is possible to use time=2015-Q1&time=2015-Q2 style in the query URL, but this makes it unfeasible to use the colon operator and requires a lot of manual typing.

Because of this, it is useful to know about other time parameters as well:

- untilTimePeriod: return dataset items from the oldest record up until the set time, for example "all data until 2000": untilTimePeriod = 2000
- sinceTimePeriod: return dataset items starting from set time, for example "all datastarting from 2008": sinceTimePeriod = 2008
- lastTimePeriod: starting from the most recent time period, how many preceding time periods should be returned? For example 10 most recent observations: lastTimePeriod = 10

Using both untilTimePeriod and sinceTimePeriod parameters in the same query is allowed, making the usage of the R colon operator unnecessary. In the case of quarterly data, using untilTimePeriod and sinceTimePeriod parameters also works, as opposed to the colon operator, so it is generally safer to use them as well.

**Other dimensions:**

In get_eurostat_json() examples nama_10_gdp dataset is filtered with two additional filter parameters:

- na_item = "B1GQ"
- unit = "CLV_I10"

Filters like these are most likely unique to the nama_10_gdp dataset (or other datasets within the same domain) and should not be used with others dataset without user discretion. By using label_eurostat() we know that "B1GQ" stands for "Gross domestic product at market prices" and "CLV_I10" means "Chain linked volumes, index 2010=100".

Different dimension codes can be translated to a natural language by using the get_eurostat_dic() function, which returns labels for individual dimension items such as na_item and unit, as opposed to label_eurostat() which does it for whole datasets. For example, the parameter na_item stands for "National accounts indicator (ESA 2010)" and unit stands for "Unit of measure".

**Language:**

All datasets have metadata available in English, French and German. If no parameter is given, the labels are returned in English.

Example:

- `lang = "fr"`

**More information:**

For more information about data filtering see Eurostat documentation on API Statistics: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+data+query#APIStatisticsdataquery-Thepara

## Citing Eurostat data

For citing datasets, use `get_bibentry()` to build a bibliography that is suitable for your reference manager of choice.

When using Eurostat data in other contexts than academic publications that in-text citations or footnotes/endnotes, the following guidelines may be helpful:

- The origin of the data should always be mentioned as "Source: Eurostat".
- The online dataset codes(s) should also be provided in order to ensure transparency and facilitate access to the Eurostat data and related methodological information. For example: "Source: Eurostat (online data code: namq_10_gdp)"
- Online publications (e.g. web pages, PDF) should include a clickable link to the dataset using the bookmark functionality available in the Eurostat data browser.

It should be avoided to associate different entities (e.g. Eurostat, National Statistical Offices, other data providers) to the same dataset or indicator without specifying the role of each of them in the treatment of data.

See also section "Eurostat: Copyright notice and free re-use of data" in `get_eurostat()` documentation.

## Disclaimer: Availability of filtering functionalities

Currently it only possible to download filtered data through API Statistics (JSON API) when using `eurostat` package, although technically filtering datasets downloaded through the SDMX Dissemination API is also supported by Eurostat. We may support this feature in the future. In the meantime, if you are interested in filtering Dissemination API data queries manually, please consult the following Eurostat documentation: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+filtering

## Strategies for handling large datasets more efficiently

Most Eurostat datasets are relatively manageable, at least on a machine with 16 GB of RAM. The largest dataset in Eurostat database, at the time of writing this, had 148362539 (148 million) values, which results in an object with 148 million rows in tidy data (long) format. The test machine with 16 GB of RAM was able to handle the second largest dataset in the database with 91 million values (rows).

There are still some methods to make data fetching functions perform faster:

- turn caching off: `get_eurostat(cache = FALSE)`
- turn cache compression off (may result in rather large cache files!): `get_eurostat(compress_file = FALSE)`

- if you want faster caching with manageable file sizes, use stringsAsFactors: `get_eurostat(cache = TRUE, compress_file = TRUE, stringsAsFactors = TRUE)`
- Use faster data.table functions: `get_eurostat(use.data.table = TRUE)`
- Keep column processing to a minimum: `get_eurostat(time_format = "raw", type = "code")` etc.
- Read `get_eurostat()` function documentation carefully so you understand what different arguments do
- Filter the dataset so that you fetch only the parts you need!

## Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari, Markus Kainu and Pyry Kantanen

## References

See `citation("eurostat")`:

```
Kindly cite the eurostat R package as follows:

  Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
  analysis of Eurostat open data with the eurostat package. The R
  Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019

  Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
  and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
  [Computer software]. R package version 4.0.0.
  https://github.com/rOpenGov/eurostat

To see these entries in BibTeX format, use 'print(<citation>,
bibtex=TRUE)', 'toBibtex(.)', or set
'options(citation.bibtex.max=999)'.
```

When citing data downloaded from Eurostat, see section "Citing Eurostat data" in `get_eurostat()` documentation.

## See Also

`search_eurostat()`, `label_eurostat()`

## Examples

```
## Not run:
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)

dd <- get_eurostat("nama_10_gdp",
  filters = list(
    geo = "FI",
    na_item = "B1GQ",
    unit = "CLV_I10"
  )
)

# A dataset with multiple time series in one
dd2 <- get_eurostat("AVIA_GOR_ME",
  select_time = c("A", "M", "Q"),
  time_format = "date_last"
)

# An example of downloading whole dataset from JSON API
dd3 <- get_eurostat("AVIA_GOR_ME",
  filters = list()
)

# Filtering a dataset from a local file
dd3_filter <- get_eurostat("AVIA_GOR_ME",
  filters = list(
    tra_meas = "FRM_BRD"
  )
)


## 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. Updated by Pyry Kantanen to support XML codelists.

## References

See citation("eurostat"):

```
# Kindly cite the eurostat R package as follows:
#
#   Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
#   analysis of Eurostat open data with the eurostat package. The R
#   Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019
#
#   Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
#   and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
#   [Computer software]. R package version 4.0.0.
#   https://github.com/rOpenGov/eurostat
#
# To see these entries in BibTeX format, use 'print(<citation>,
# bibtex=TRUE)', 'toBibtex(.)', or set
# 'options(citation.bibtex.max=999)'.
```

## 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_folder` *Get all datasets in a folder*

## Description

Loops over all files in a Eurostat database folder, downloads the data and assigns the datasets to environment.

## Usage

```
get_eurostat_folder(code, env = .EurostatEnv)
```

## Arguments

| | |
|---|---|
| code | Folder code from Eurostat Table of Contents. |
| env | Name of the environment where downloaded datasets are assigned. Default is .EurostatEnv. If NULL, datasets are returned as a list object. |

## Details

The datasets are assigned into .EurostatEnv by default, using dataset codes as object names. The datasets are downloaded from SDMX API as TSV files, meaning that they are returned without filtering. No filters can be provided using this function.

Please do not attempt to download too many datasets or the whole database at once. The number of datasets that can be downloaded at once is hardcoded to 20. The function also asks the user for confirmation if the number of datasets in a folder is more than 10. This is by design to discourage straining Eurostat API.

## Data source: Eurostat Table of Contents

The Eurostat Table of Contents (TOC) is downloaded from https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=en (default) or from French or German language variants: https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=fr https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=de

See Eurostat documentation on TOC items: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+-+Detailed+guidelines+-+Catalogue+API+-+TOC

## Data source: Eurostat SDMX 2.1 Dissemination API

Data is downloaded from Eurostat SDMX 2.1 API endpoint as compressed TSV files that are transformed into tabular format. See Eurostat documentation for more information: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+query

The new dissemination API replaces the old bulk download facility that was used by Eurostat before October 2023 and by the eurostat R package versions before 4.0.0. See Eurostat documentation about the transition from Bulk Download to API for more information about the differences between

the old bulk download facility and the data provided by the new API connection: https://wikis.ec.europa.eu/display/EUROSTATHELP/Transition+-+from+Eurostat+Bulk+Download+to+API

See especially the document Migrating_to_API_TSV.pdf that describes the changes in TSV file format in new applications.

For more information about SDMX 2.1, see SDMX standards: Section 7: Guidelines for the use of web services, Version 2.1: https://sdmx.org/wp-content/uploads/SDMX_2-1_SECTION_7_WebServicesGuidelines.pdf

## Author(s)

Pyry Kantanen

## See Also

get_eurostat_toc() toc_count_children() toc_determine_hierarchy() toc_list_children() toc_count_whitespace()

---

# get_eurostat_geospatial

*Download Geospatial Data from GISCO*

---

## Description

Downloads either a simple features (sf) or a data_frame of NUTS regions. This function is a wrapper of giscoR::gisco_get_nuts(). This function requires to have installed the packages **sf** and **giscoR**.

## 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 = "DEPRECATED",
  ...
)
```

## Arguments

| | |
|---|---|
| output_class | Class of object returned, either `sf` `simple features` or `df` (`data_frame`). `spdf` output has been soft-deprecated, the function would switch to `sf`. |
| resolution | Resolution of the geospatial data. One of<br>• "60" (1:60million),<br>• "20" (1:20million)<br>• "10" (1:10million)<br>• "03" (1:3million) or<br>• "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`. |
| 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. See `set_eurostat_cache_dir()`. If `NULL` and the cache dir has not been set globally the file would be stored in the `tempdir()`. |
| crs | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA<br>• "3857" - Pseudo-Mercator |
| make_valid | Deprecated |
| ... | Arguments passed on to `giscoR::gisco_get_nuts`<br>`verbose` Logical, displays information. Useful for debugging, default is `FALSE`.<br>`spatialtype` Type of geometry to be returned:<br>• **"BN"**: Boundaries - `LINESTRING` object.<br>• **"LB"**: Labels - `POINT` object.<br>• **"RG"**: Regions - `MULTIPOLYGON/POLYGON` object.<br>`country` Optional. A character vector of country codes. It could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes. Mixed types (as `c("Turkey","US","FRA")`) would not work. See also `countrycode::countrycode()`.<br>`nuts_id` Optional. A character vector of NUTS IDs. |

## Details

The objects downloaded from GISCO should contain all or some of the following variable columns:

- **id**: JSON id code, the same as **NUTS_ID**. See **NUTS_ID** below for further clarification.
- **LEVL_CODE**: NUTS level code: 0 (national level), 1 (major socio-economic regions), 2 (basic regions for the application of regional policies) or 3 (small regions).
- **NUTS_ID**: NUTS ID code, consisting of country code and numbers (1 for NUTS 1, 2 for NUTS 2 and 3 for NUTS 3)

- **CNTR_CODE**: Country code: two-letter ISO code (ISO 3166 alpha-2), except in the case of Greece (EL).
- **NAME_LATN**: NUTS name in local language, transliterated to Latin script
- **NUTS_NAME**: NUTS name in local language, in local script.
- **MOUNT_TYPE**: Mountain typology for NUTS 3 regions.
  - 1: "where more than 50 % of the surface is covered by topographic mountain areas"
  - 2: "in which more than 50 % of the regional population lives in topographic mountain areas"
  - 3: "where more than 50 % of the surface is covered by topographic mountain areas and where more than 50 % of the regional population lives in these mountain areas"
  - 4: non-mountain region / other region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 and non-EU countries)
- **URBN_TYPE**: Urban-rural typology for NUTS 3 regions.
  - 1: predominantly urban region
  - 2: intermediate region
  - 3: predominantly rural region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 regions)
- **COAST_TYPE**: Coastal typology for NUTS 3 regions.
  - 1: coastal (on coast)
  - 2: coastal (>= 50% of population living within 50km of the coastline)
  - 3: non-coastal region
  - 0: no classification provided (e.g. in the case of NUTS 1 and NUTS 2 regions)
- **FID**: Same as NUTS_ID.
- **geo**: Same as NUTS_ID, added for for easier joins with dplyr. Consider the status of this column "questioning" and use other columns for joins when possible.
- **geometry**: geospatial information.

## Value

a sf or data_frame

## Eurostat: Copyright notice and free re-use of data

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: https://ec.europa.eu/eurostat/about-us/policies/copyright

"(c) European Union, 1995 - today

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat;

- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information."

For exceptions to the abovementioned principles see Eurostat website

**Data source: GISCO - General Copyright**

"Eurostat's general copyright notice and licence policy is applicable and can be consulted here: https://ec.europa.eu/eurostat/about-us/policies/copyright

Please also be aware of the European Commission's general conditions: https://commission.europa.eu/legal-notice_en

Moreover, there are specific provisions applicable to some of the following datasets available for downloading. The download and usage of these data is subject to their acceptance:

- Administrative Units / Statistical Units
- Population distribution / Demography
- Transport Networks
- Land Cover
- Elevation (DEM)"

Of the abovementioned datasets, Administrative Units / Statistical Units is applicable if the user wants to draw maps with borders provided by GISCO / EuroGeographics.

**Data source: GISCO - Administrative Units / Statistical Units**

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the GISCO website: GISCO: Geographical information and maps - Administrative units/statistical units

"In addition to the general copyright and licence policy applicable to the whole Eurostat website, the following specific provisions apply to the datasets you are downloading. The download and usage of these data is subject to the acceptance of the following clauses:

1. The Commission agrees to grant the non-exclusive and not transferable right to use and process the Eurostat/GISCO geographical data downloaded from this page (the "data").
2. The permission to use the data is granted on condition that:
   (a) the data will not be used for commercial purposes;
   (b) the source will be acknowledged. A copyright notice, as specified below, will have to be visible on any printed or electronic publication using the data downloaded from this page."

**Copyright notice:**

When data downloaded from this page is used in any printed or electronic publication, in addition to any other provisions applicable to the whole Eurostat website, data source will have to be acknowledged in the legend of the map and in the introductory page of the publication with the following copyright notice:

EN: © EuroGeographics for the administrative boundaries

FR: © EuroGeographics pour les limites administratives

DE: © EuroGeographics bezüglich der Verwaltungsgrenzen

For publications in languages other than English, French or German, the translation of the copyright notice in the language of the publication shall be used.

If you intend to use the data commercially, please contact EuroGeographics for information regarding their licence agreements."

## Author(s)

Markus Kainu markuskainu@gmail.com, Diego Hernangomez https://github.com/dieghernan/

## Source

Data source: Eurostat

© EuroGeographics for the administrative boundaries

Data downloaded using **giscoR**

## See Also

giscoR::gisco_get_nuts()

Other geospatial: eurostat_geodata_60_2016

## Examples

```
# Uses cached dataset
sf <- get_eurostat_geospatial(
  output_class = "sf",
  resolution = "60",
  nuts_level = "all"
)
# Downloads dataset from server
sf2 <- get_eurostat_geospatial(
  output_class = "sf",
  resolution = "20",
  nuts_level = "all"
)
df <- get_eurostat_geospatial(
  output_class = "df",
  nuts_level = "0"
)
```

## get_eurostat_interactive *Get Eurostat data interactive*

### Description

A simple interactive helper function to go through the steps of downloading and/or finding suitable eurostat datasets.

### Usage

```
get_eurostat_interactive(code = NULL)
```

### Arguments

| | |
|---|---|
| code | A unique identifier / code for the dataset of interest. If code is not known search_eurostat() function can be used to search Eurostat table of contents. |

### Details

This function is intended to enable easy exploration of different eurostat package functionalities and functions. In order to not drown the end user in endless menus this function does not allow for setting all possible get_eurostat() function arguments. It is possible to set time_format, type, lang, stringsAsFactors, keepFlags, and use.data.table in the interactive menus.

In some datasets setting these parameters may result in a "Error in label_eurostat" error, for example: "labels for XXXXXX includes duplicated labels in the Eurostat dictionary". In these cases, and with other more complex queries, please use get_eurostat() function directly.

### See Also

get_eurostat()

## 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 = "code",
  lang = "en",
  stringsAsFactors = FALSE,
  proxy = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| id | A unique identifier / code for the dataset of interest. If code is not known search_eurostat() function can be used to search Eurostat table of contents. |
| 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 information on filters and limitations per query. |
| type | A type of variables, "code" (default), "label" or "both". The parameter "both" will return a data_frame with named vectors, labels as values and codes as names. |
| lang | 2-letter language code, default is "en" (English), other options are "fr" (French) and "de" (German). Used for labeling datasets. |
| stringsAsFactors | if TRUE (the default) variables are converted to factors in the original Eurostat order. If FALSE they are returned as strings. |
| proxy | Use proxy, TRUE or FALSE (default). |
| ... | Arguments passed on to httr2::req_proxy<br>req A request.<br>url,port Location of proxy.<br>username,password Login details for proxy, if needed.<br>auth Type of HTTP authentication to use. Should be one of the following: basic, digest, digest_ie, gssnegotiate, ntlm, any. |

## 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. See "Filtering datasets" section below for more detailed information about filters.

To use a proxy to connect, proxy arguments can be passed to httr2::req_perform() via httr2::req_proxy() - see latter function documentation for parameter names that can be passed with .... A non-functional example: get_eurostat_json(id, filters, proxy = TRUE, url = "127.0.0.1", port = 80).

When retrieving data from Eurostat JSON API the user may encounter errors. For end user convenience, we have provided a ready-made internal dataset `sdmx_http_errors` that contains descriptive labels and descriptions about the possible interpretation or cause of each error. These messages are returned if the API returns a status indicating a HTTP error (400 or greater).

The Eurostat implementation seems to be based on SDMX 2.1, which is the reason we've used SDMX Standards guidelines as a supplementary source that we have included in the dataset. What this means in practice is that the dataset contains error codes and their mappings that are not mentioned in the Eurostat website. We hope you never encounter them.

## Value

A dataset as an object of `data.frame` class.

## Data source: Eurostat API Statistics (JSON API)

Data is downloaded from Eurostat API Statistics. See Eurostat documentation for more information about data queries in API Statistics https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+data+query

This replaces the old JSON Web Services that was used by Eurostat before February 2023 and by the eurostat R package versions before 3.7.13. See Eurostat documentation about the migration from JSON web service to API Statistics for more information about the differences between the old and the new service: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+migrating+from+JSON+web+service+to+API+Statistics

For easily viewing which filtering options are available - in addition to the default ones, time and language - Eurostat Web services Query builder tool may be useful: https://ec.europa.eu/eurostat/web/query-builder

## Filtering datasets

When using Eurostat API Statistics (JSON API), datasets can be filtered before they are downloaded and saved in local memory. The general format for filter parameters is `<DIMENSION_CODE>=<VALUE>`.

Filter parameters are optional but the used dimension codes must be present in the data product that is being queried. Dimension codes can vary between different data products so it may be useful to examine new datasets in Eurostat data browser beforehand. However, most if not all Eurostat datasets concern European countries and contain information that was gathered at some point in time, so `geo` and `time` dimension codes can usually be used.
`<DIMENSION_CODE>` and `<VALUE>` are case-insensitive and they can be written in lowercase or uppercase in the query.

Parameters are passed onto the `eurostat` package functions `get_eurostat()` and `get_eurostat_json()` as a list item. If an individual item contains multiple items, as it often can be in the case of `geo` parameters and other optional items, they must be in the form of a vector: `c("FI", "SE")`. For examples on how to use these parameters, see function examples below.

### Time parameters:

`time` and `time_period` address the same `TIME_PERIOD` dimension in the dataset and can be used interchangeably. In the Eurostat documentation it is stated that "Using more than one Time parameter in the same query is not accepted", but practice has shown that actually Eurostat API allows

multiple time parameters in the same query. This makes it possible to use R colon operator when writing queries, so time = c(2015:2018) translates to &time=2015&time=2016&time=2017&time=2018.

The only exception to this is when the queried dataset contains e.g. quarterly data and TIME_PERIOD is saved as 2015-Q1, 2015-Q2 etc. Then it is possible to use time=2015-Q1&time=2015-Q2 style in the query URL, but this makes it unfeasible to use the colon operator and requires a lot of manual typing.

Because of this, it is useful to know about other time parameters as well:

- untilTimePeriod: return dataset items from the oldest record up until the set time, for example "all data until 2000": untilTimePeriod = 2000
- sinceTimePeriod: return dataset items starting from set time, for example "all datastarting from 2008": sinceTimePeriod = 2008
- lastTimePeriod: starting from the most recent time period, how many preceding time periods should be returned? For example 10 most recent observations: lastTimePeriod = 10

Using both untilTimePeriod and sinceTimePeriod parameters in the same query is allowed, making the usage of the R colon operator unnecessary. In the case of quarterly data, using untilTimePeriod and sinceTimePeriod parameters also works, as opposed to the colon operator, so it is generally safer to use them as well.

**Other dimensions:**

In get_eurostat_json() examples nama_10_gdp dataset is filtered with two additional filter parameters:

- na_item = "B1GQ"
- unit = "CLV_I10"

Filters like these are most likely unique to the nama_10_gdp dataset (or other datasets within the same domain) and should not be used with others dataset without user discretion. By using label_eurostat() we know that "B1GQ" stands for "Gross domestic product at market prices" and "CLV_I10" means "Chain linked volumes, index 2010=100".

Different dimension codes can be translated to a natural language by using the get_eurostat_dic() function, which returns labels for individual dimension items such as na_item and unit, as opposed to label_eurostat() which does it for whole datasets. For example, the parameter na_item stands for "National accounts indicator (ESA 2010)" and unit stands for "Unit of measure".

**Language:**

All datasets have metadata available in English, French and German. If no parameter is given, the labels are returned in English.

Example:

- lang = "fr"

**More information:**

For more information about data filtering see Eurostat documentation on API Statistics: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+Statistics+-+data+query#APIStatisticsdataquery-Thepara

### Eurostat: Copyright notice and free re-use of data

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: https://ec.europa.eu/eurostat/about-us/policies/copyright

"(c) European Union, 1995 - today

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat;
- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information."

For exceptions to the abovementioned principles see Eurostat website

### Citing Eurostat data

For citing datasets, use `get_bibentry()` to build a bibliography that is suitable for your reference manager of choice.

When using Eurostat data in other contexts than academic publications that in-text citations or footnotes/endnotes, the following guidelines may be helpful:

- The origin of the data should always be mentioned as "Source: Eurostat".
- The online dataset codes(s) should also be provided in order to ensure transparency and facilitate access to the Eurostat data and related methodological information. For example: "Source: Eurostat (online data code: namq_10_gdp)"
- Online publications (e.g. web pages, PDF) should include a clickable link to the dataset using the bookmark functionality available in the Eurostat data browser.

It should be avoided to associate different entities (e.g. Eurostat, National Statistical Offices, other data providers) to the same dataset or indicator without specifying the role of each of them in the treatment of data.

See also section "Eurostat: Copyright notice and free re-use of data" in `get_eurostat()` documentation.

### Disclaimer: Availability of filtering functionalities

Currently it only possible to download filtered data through API Statistics (JSON API) when using `eurostat` package, although technically filtering datasets downloaded through the SDMX Dissemination API is also supported by Eurostat. We may support this feature in the future. In the meantime, if you are interested in filtering Dissemination API data queries manually, please consult the following Eurostat documentation: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+filtering

### Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari Markus Kainu and Pyry Kantanen

## References

See `citation("eurostat")`:

```
Kindly cite the eurostat R package as follows:

  Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
  analysis of Eurostat open data with the eurostat package. The R
  Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019

  Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
  and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
  [Computer software]. R package version 4.0.0.
  https://github.com/rOpenGov/eurostat

To see these entries in BibTeX format, use 'print(<citation>,
bibtex=TRUE)', 'toBibtex(.)', or set
'options(citation.bibtex.max=999)'.
```

When citing data downloaded from Eurostat, see section "Citing Eurostat data" in `get_eurostat()` documentation.

## See Also

`httr2::req_proxy()`

## Examples

```
## Not run:
# Generally speaking these queries would be done through get_eurostat
tmp <- get_eurostat_json("nama_10_gdp")
yy <- get_eurostat_json("nama_10_gdp", filters = list(
  geo = c("FI", "SE", "EU28"),
  time = c(2015:2023),
  lang = "FR",
  na_item = "B1GQ",
  unit = "CLV_I10"
))

# TIME_PERIOD filter works also with the new JSON API
yy2 <- get_eurostat_json("nama_10_gdp", filters = list(
   geo = c("FI", "SE", "EU28"),
   TIME_PERIOD = c(2015:2023),
   lang = "FR",
   na_item = "B1GQ",
   unit = "CLV_I10"
))

# An example from get_eurostat
dd <- get_eurostat("nama_10_gdp",
  filters = list(
```

```
  geo = "FI",
  na_item = "B1GQ",
  unit = "CLV_I10"
))

## End(Not run)
```

---

`get_eurostat_raw` *Download Data from Eurostat Dissemination API*

---

## Description

Download data from the eurostat database through the new dissemination API.

## Usage

```
get_eurostat_raw(id, use.data.table = FALSE)
```

## Arguments

| | |
|---|---|
| `id` | A unique identifier / code for the dataset of interest. If code is not known `search_eurostat()` function can be used to search Eurostat table of contents. |
| `use.data.table` | Use faster data.table functions? Default is FALSE. On Windows requires that RTools is installed. |

## 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.

## Data source: Eurostat SDMX 2.1 Dissemination API

Data is downloaded from Eurostat SDMX 2.1 API endpoint as compressed TSV files that are transformed into tabular format. See Eurostat documentation for more information: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+query

The new dissemination API replaces the old bulk download facility that was used by Eurostat before October 2023 and by the eurostat R package versions before 4.0.0. See Eurostat documentation about the transition from Bulk Download to API for more information about the differences between the old bulk download facility and the data provided by the new API connection: https://wikis.ec.europa.eu/display/EUROSTATHELP/Transition+-+from+Eurostat+Bulk+Download+to+API

See especially the document Migrating_to_API_TSV.pdf that describes the changes in TSV file format in new applications.

For more information about SDMX 2.1, see SDMX standards: Section 7: Guidelines for the use of web services, Version 2.1: https://sdmx.org/wp-content/uploads/SDMX_2-1_SECTION_7_WebServicesGuidelines.pdf

### Eurostat: Copyright notice and free re-use of data

The following copyright notice is provided for end user convenience. Please check up-to-date copyright information from the eurostat website: https://ec.europa.eu/eurostat/about-us/policies/copyright

"(c) European Union, 1995 - today

Eurostat has a policy of encouraging free re-use of its data, both for non-commercial and commercial purposes. All statistical data, metadata, content of web pages or other dissemination tools, official publications and other documents published on its website, with the exceptions listed below, can be reused without any payment or written licence provided that:

- the source is indicated as Eurostat;
- when re-use involves modifications to the data or text, this must be stated clearly to the end user of the information."

For exceptions to the abovementioned principles see Eurostat website

### Citing Eurostat data

For citing datasets, use `get_bibentry()` to build a bibliography that is suitable for your reference manager of choice.

When using Eurostat data in other contexts than academic publications that in-text citations or footnotes/endnotes, the following guidelines may be helpful:

- The origin of the data should always be mentioned as "Source: Eurostat".
- The online dataset codes(s) should also be provided in order to ensure transparency and facilitate access to the Eurostat data and related methodological information. For example: "Source: Eurostat (online data code: namq_10_gdp)"
- Online publications (e.g. web pages, PDF) should include a clickable link to the dataset using the bookmark functionality available in the Eurostat data browser.

It should be avoided to associate different entities (e.g. Eurostat, National Statistical Offices, other data providers) to the same dataset or indicator without specifying the role of each of them in the treatment of data.

See also section "Eurostat: Copyright notice and free re-use of data" in `get_eurostat()` documentation.

### Disclaimer: Availability of filtering functionalities

Currently it only possible to download filtered data through API Statistics (JSON API) when using `eurostat` package, although technically filtering datasets downloaded through the SDMX Dissemination API is also supported by Eurostat. We may support this feature in the future. In the meantime, if you are interested in filtering Dissemination API data queries manually, please consult the following Eurostat documentation: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+SDMX+2.1+-+data+filtering

### Author(s)

Przemyslaw Biecek, Leo Lahti, Janne Huovari and Pyry Kantanen

## References

See `citation("eurostat")`:

```
# Kindly cite the eurostat R package as follows:
#
#   Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
#   analysis of Eurostat open data with the eurostat package. The R
#   Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019
#
#   Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
#   and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
#   [Computer software]. R package version 4.0.0.
#   https://github.com/rOpenGov/eurostat
#
# To see these entries in BibTeX format, use 'print(<citation>,
# bibtex=TRUE)', 'toBibtex(.)', or set
# 'options(citation.bibtex.max=999)'.
```

## 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(lang = "en")
```

## Arguments

| | |
|---|---|
| `lang` | 2-letter language code, default is "en" (English), other options are "fr" (French) and "de" (German). Used for labeling datasets. |

## Details

In the downloaded Eurostat Table of Contents the 'code' column values are refer to the function 'id' that is used as an argument in certain functions when downloading datasets.

## Value

A tibble with nine columns:

**title** Dataset title in English (default)

**code** Each item (dataset, table and folder) of the TOC has a unique code which allows it to be identified in the API. Used in the `get_eurostat()` and `get_eurostat_raw()` functions to retrieve datasets.

**type** dataset, folder or table

**last.update.of.data** Date, indicates the last time the dataset/table was updated (format `DD.MM.YYYY` or `%d.%m.%Y`)

**last.table.structure.change** Date, indicates the last time the dataset/table structure was modified (format `DD.MM.YYYY` or `%d.%m.%Y`)

**data.start** Date of the oldest value included in the dataset (if available) (format usually `YYYY` or `%Y` but can also be `YYYY-MM`, `YYYY-MM-DD`, `YYYY-SN`, `YYYY-QN` etc.)

**data.end** Date of the most recent value included in the dataset (if available) (format usually `YYYY` or `%Y` but can also be `YYYY-MM`, `YYYY-MM-DD`, `YYYY-SN`, `YYYY-QN` etc.)

**values** Number of actual values included in the dataset

**hierarchy** Hierarchy of the data navigation tree, represented in the original txt file by a 4-spaces indentation prefix in the title

## Data source: Eurostat Table of Contents

The Eurostat Table of Contents (TOC) is downloaded from https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=en (default) or from French or German language variants: https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=fr https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=de

See Eurostat documentation on TOC items: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+-+Detailed+guidelines+-+Catalogue+API+-+TOC

## Author(s)

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

## References

See `citation("eurostat")`:

```
Kindly cite the eurostat R package as follows:

  Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
  analysis of Eurostat open data with the eurostat package. The R
  Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019

  Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
  and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
  [Computer software]. R package version 4.0.0.
```

```
  https://github.com/rOpenGov/eurostat

To see these entries in BibTeX format, use 'print(<citation>,
bibtex=TRUE)', 'toBibtex(.)', or set
'options(citation.bibtex.max=999)'.
```

When citing data downloaded from Eurostat, see section "Citing Eurostat data" in `get_eurostat()` documentation.

## See Also

`get_eurostat()`, `search_eurostat()`

## Examples

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

# Convert columns containing dates as character into Date class
# Last update of data
tmp[[4]] <- as.Date(tmp[[4]], format = c("%d.%m.%Y"))
# Last table structure change
tmp[[5]] <- as.Date(tmp[[5]], format = c("%d.%m.%Y"))
# Data start, contains several formats (date, week, month quarter, semester)
# Unfortunately semesters are not directly supported so they need to be
# changed into quarters
tmp$data.start <- gsub("S2", "Q3", tmp$data.start)
tmp$data.start <- lubridate::as_date(
 x = tmp$data.start,
 format = c("%Y", "%Y-Q%q", "%Y-W%W", "%Y-S%q", "%Y-%m-%d", "%Y-%m")
 )
# Data end, same as data start
tmp$data.end <- gsub("S2", "Q3", tmp$data.end)
tmp$data.end <- lubridate::as_date(
 x = tmp$data.end,
 format = c("%Y", "%Y-Q%q", "%Y-W%W", "%Y-S%q", "%Y-%m-%d", "%Y-%m")
 )
```

---

`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 for data downloaded from new dissemination API*

---

### 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 = NULL, id, 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 | 2-letter language code, default is "en" (English), other options are "fr" (French) and "de" (German). Used for labeling datasets. |
| 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. |
| id | A unique identifier / code for the dataset of interest. If code is not known search_eurostat() function can be used to search Eurostat table of contents. |

## 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.
- `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")
# Note that the dataset id must be provided in label_eurostat_vars
label_eurostat_vars(id = "nama_10_lp_ulc", x = "geo", lang = "en")
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)
```

## list_eurostat_cache_items *Output cache information as data.frame*

### Description

Parses cache_list.json file and returns a data.frame

### Usage

```
list_eurostat_cache_items(cache_dir = NULL)
```

### Arguments

| | |
|---|---|
| cache_dir | a path to a cache directory. NULL (default) uses and creates 'eurostat' directory in the temporary directory defined by base R tempdir() function. The user can set the cache directory to an existing directory by using this argument. The cache directory can also be set with set_eurostat_cache_dir() function. |

### Value

A data.frame object with 3 columns: dataset code, download date and query md5 hash

## search_eurostat *Grep Datasets Titles from Eurostat*

### Description

Lists datasets from eurostat table of contents with the particular pattern in item titles.

### Usage

```
search_eurostat(
  pattern,
  type = "dataset",
  column = "title",
  fixed = TRUE,
  lang = "en"
)
```

## Arguments

| | |
|---|---|
| pattern | Text string that is used to search from dataset, folder or table titles, depending on the type argument. |
| type | Selection for types of datasets to be searched. Default is dataset, other possible options are table, folder and all for all types. |
| column | Selection for the column of TOC where search is done. Default is title, other possible option is code. |
| fixed | logical. If TRUE (default), pattern is a string to be matched as is. See grep() documentation for more information. |
| lang | 2-letter language code, default is "en" (English), other options are "fr" (French) and "de" (German). Used for labeling datasets. |

## 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.

If you wish to perform searches on other fields than item title, you can download the Eurostat Table of Contents manually using get_eurostat_toc() function and use grep() function normally. The data browser on Eurostat website may also return useful results.

## Value

A tibble with nine columns:

**title** Dataset title in English (default)

**code** Each item (dataset, table and folder) of the TOC has a unique code which allows it to be identified in the API. Used in the get_eurostat() and get_eurostat_raw() functions to retrieve datasets.

**type** dataset, folder or table

**last.update.of.data** Date, indicates the last time the dataset/table was updated (format DD.MM.YYYY or %d.%m.%Y)

**last.table.structure.change** Date, indicates the last time the dataset/table structure was modified (format DD.MM.YYYY or %d.%m.%Y)

**data.start** Date of the oldest value included in the dataset (if available) (format usually YYYY or %Y but can also be YYYY-MM, YYYY-MM-DD, YYYY-SN, YYYY-QN etc.)

**data.end** Date of the most recent value included in the dataset (if available) (format usually YYYY or %Y but can also be YYYY-MM, YYYY-MM-DD, YYYY-SN, YYYY-QN etc.)

**values** Number of actual values included in the dataset

**hierarchy** Hierarchy of the data navigation tree, represented in the original txt file by a 4-spaces indentation prefix in the title

### Data source: Eurostat Table of Contents

The Eurostat Table of Contents (TOC) is downloaded from https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=en (default) or from French or German language variants: https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=fr https://ec.europa.eu/eurostat/api/dissemination/catalogue/toc/txt?lang=de

See Eurostat documentation on TOC items: https://wikis.ec.europa.eu/display/EUROSTATHELP/API+-+Detailed+guidelines+-+Catalogue+API+-+TOC

### Author(s)

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

### References

See `citation("eurostat")`:

```
Kindly cite the eurostat R package as follows:

  Lahti L., Huovari J., Kainu M., and Biecek P. (2017). Retrieval and
  analysis of Eurostat open data with the eurostat package. The R
  Journal 9(1), pp. 385-392. doi: 10.32614/RJ-2017-019

  Lahti, L., Huovari J., Kainu M., Biecek P., Hernangomez D., Antal D.,
  and Kantanen P. (2023). eurostat: Tools for Eurostat Open Data
  [Computer software]. R package version 4.0.0.
  https://github.com/rOpenGov/eurostat

To see these entries in BibTeX format, use 'print(<citation>,
bibtex=TRUE)', 'toBibtex(.)', or set
'options(citation.bibtex.max=999)'.
```

When citing data downloaded from Eurostat, see section "Citing Eurostat data" in `get_eurostat()` documentation.

### See Also

`get_eurostat()`, `search_eurostat()`

### 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

Disposable income of private households by NUTS 2 regions Retrieved with: `tgs00026 <- get_eurostat("tgs00026", time_format = "raw")` Data retrieval date: 2022-06-27

## See Also

Other datasets: eu_countries, eurostat_geodata_60_2016

# Index