

# Package ‘ecb’

March 19, 2025

**Type** Package

**Title** Programmatic Access to the European Central Bank's Data Portal

**Version** 0.4.3

**Date** 2025-03-19

**Description** Provides an interface to the European Central Bank's Data Portal API, allowing for programmatic retrieval of a vast quantity of statistical data.

**License** CC0

**URL** <https://github.com/expersso/ecb>

**BugReports** <https://github.com/expersso/ecb/issues>

**Suggests** knitr, rmarkdown, dplyr, lubridate, ggplot2, testthat, zoo

**Imports** curl, rsdmx, xml2, httr

**VignetteBuilder** knitr

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Eric Persson [aut, cre],  
Franz X. Mohr [ctb] (<<https://orcid.org/0009-0003-8890-7781>>)

**Maintainer** Eric Persson <expersso5@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-03-19 11:30:02 UTC

## Contents

convert_dates . . . . .	2
get_data . . . . .	2
get_dataflows . . . . .	4
get_description . . . . .	4
get_dimensions . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

convert_dates	<i>Format date variable retrieved from the SDW into a proper date variable</i>
---------------	--

---

**Description**

Format date variable retrieved from the SDW into a proper date variable

**Usage**

```
convert_dates(x)
```

**Arguments**

x	A vector of dates
---	-------------------

**Value**

A date-formatted vector

**Examples**

```
hicip <- get_data("ICP.M.U2.N.000000.4.ANR")
hicip$obstime <- convert_dates(hicip$obstime)
str(hicip)
```

---

get_data	<i>Retrieve data from the ECB Statistical Data Warehouse API</i>
----------	--

---

**Description**

Retrieve data from the ECB Statistical Data Warehouse API

**Usage**

```
get_data(key, filter = NULL, ...)
```

**Arguments**

key	A character string identifying the series to be retrieved
filter	A named list with additional filters (see details)
...	Arguments passed to GET (e.g. timeout(10) to add maximum request time)

## Details

The filter option of `get_data()` takes a named list of key-value pairs. If left blank, it returns all data for the current version.

Available filter parameters:

- startPeriod & endPeriod
  - YYYY for annual data (e.g.: 2013)
  - YYYY-S[1-2] for semi-annual data (e.g.: 2013-S1)
  - YYYY-Q[1-4] for quarterly data (e.g.: 2013-Q1)
  - YYYY-MM for monthly data (e.g.: 2013-01)
  - YYYY-W[01-53] for weekly data (e.g.: 2013-W01)
  - YYYY-MM-DD for daily data (e.g.: 2013-01-01)
- updatedAfter
  - A timestamp to retrieve the latest version of changed values in the database since a certain point in time
  - Example: `filter = list(updatedAfter = 2009-05-15T14:15:00+01:00)`
- firstNObservations & lastNObservations
  - Example: `filter = list(firstNObservations = 12)` retrieves the first 12 observations of all specified series
- detail
  - Possible options: `full/dataonly/serieskeyonly/nodata`
  - `dataonly` is the default
  - Use `serieskeyonly` or `nodata` to list series that match a certain query, without returning the actual data
  - An alternative to using `serieskeys/nodata` is the convenience function `get_dimensions()`, which returns a list of dataframes with dimensions and explanations (see extended example below).
  - `full` returns both the series values and all metadata. This entails retrieving much more data than with the 'dataonly' option.
- includeHistory (not currently implemented)
  - `false` (default) returns only version currently in production
  - `true` returns version currently in production, as well as all previous versions

See the [SDW API](#) for more details.

## Value

A data frame

## Examples

```
# Get monthly data on annualized euro area headline HICP
hicip <- get_data("ICP.M.U2.N.000000.4.ANR")
head(hicip)
```

---

get_dataflows	<i>Retrieve data frame of all datasets in the ECB Statistical Data Warehouse</i>
---------------	--

---

**Description**

Retrieve data frame of all datasets in the ECB Statistical Data Warehouse

**Usage**

```
get_dataflows(...)
```

**Arguments**

... Arguments passed to GET (e.g. timeout(10) to add maximum request time)

**Value**

A dataframe

**Examples**

```
df <- get_dataflows()
head(df)
```

---

get_description	<i>Get full, human-readable description of a series</i>
-----------------	---

---

**Description**

Get full, human-readable description of a series

**Usage**

```
get_description(key)
```

**Arguments**

key A character string identifying the series to be retrieved

**Value**

A character vector

**Examples**

```
get_description("ICP.M.DE.N.000000+XEF000.4.ANR")
```

---

get_dimensions	<i>Retrieve dimensions of series in the ECB's SDW</i>
----------------	---

---

**Description**

Retrieve dimensions of series in the ECB's SDW

**Usage**

```
get_dimensions(key, ...)
```

**Arguments**

key	A character string identifying the series to be retrieved
...	Arguments passed to GET (e.g. timeout(10) to add maximum request time)

**Value**

A list of data frames, one for each series retrieved

**Examples**

```
h1cp_dims <- get_dimensions("ICP.M.U2.N.000000.4.ANR")  
h1cp_dims[[1]]
```

# Index

`convert_dates`, [2](#)

`get_data`, [2](#)

`get_dataflows`, [4](#)

`get_description`, [4](#)

`get_dimensions`, [5](#)