

Package ‘bbk’

May 7, 2026

Title Client for Central Bank APIs

Version 0.9.0

Description A client for retrieving data and metadata from major central bank APIs. It supports access to the 'Bundesbank SDMX Web Service API' (<<https://www.bundesbank.de/en/statistics/time-series-databases/help-for-sdmx-web-service/web-service-interface-data>>), the 'Swiss National Bank Data Portal' (<<https://data.snb.ch/en>>), the 'European Central Bank Data Portal API' (<<https://data.ecb.europa.eu/help/api/overview>>), the 'Bank of England Interactive Statistical Database' (<<https://www.bankofengland.co.uk/boeapps/database>>), the 'Banco de España API' (<<https://www.bde.es/webbe/en/estadisticas/recursos/api-estadisticas-bde.html>>), the 'Bank for International Settlements SDMX Web Service' (<<https://stats.bis.org/api-doc/v1/>>), the 'Banque de France Web Service' (<<https://webstat.banque-france.fr/en/pages/guide-migration-api/>>), the 'Norges Bank SDMX Web Service' (<<https://www.norges-bank.no/en/topics/Statistics/open-data/>>), the 'Oesterreichische Nationalbank Web Service' (<<https://www.oenb.at/en/Statistics/User-Defined-Tables/webservice.html>>), and 'Bank of Canada Valet API' (<<https://www.bankofcanada.ca/valet/docs>>).

License MIT + file LICENSE

URL <https://m-muecke.github.io/bbk/>, <https://github.com/m-muecke/bbk>

BugReports <https://github.com/m-muecke/bbk/issues>

Depends R (>= 4.1.0)

Imports checkmate, curl, data.table (>= 1.17.0), httr2 (>= 1.0.0), jsonlite, stats, utils, xml2

Suggests ggplot2, scales, testthat (>= 3.3.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation yes

Author Maximilian Mücke [aut, cre] (ORCID:
<<https://orcid.org/0009-0000-9432-9795>>)

Maintainer Maximilian Mücke <muecke.maximilian@gmail.com>

Repository CRAN

Date/Publication 2026-03-21 12:10:02 UTC

Contents

bbk_data	3
bbk_dimension	5
bbk_metadata	6
bbk_series	7
bde_data	8
bde_latest	9
bdf_codelist	10
bdf_data	11
bdf_dataset	12
bis_data	13
bis_dimension	14
bis_metadata	15
boc_catalog	16
boc_data	17
boc_fx_rates	18
boc_metadata	19
boe_data	20
cache	21
ecb_data	22
ecb_dimension	24
ecb_fx_rates	25
ecb_metadata	26
nob_data	27
nob_dimension	28
nob_metadata	29
onb_data	30
onb_dimension	31
onb_frequency	32
onb_hierarchy	33
onb_metadata	34
onb_toc	35
snb_data	35
snb_dimension	36
srb_calendar	37
srb_cross_rates	38
srb_data	39

<i>bbk_data</i>	3
<i>srb_series</i>	40
Index	42

bbk_data *Fetch Deutsche Bundesbank (BBk) data*

Description

Retrieve time series data from the Bundesbank SDMX Web Service.

Usage

```
bbk_data(
  flow,
  key = NULL,
  start_period = NULL,
  end_period = NULL,
  first_n = NULL,
  last_n = NULL
)
```

Arguments

<i>flow</i>	(character(1)) The flow to query, 5-8 characters. See bbk_metadata() for available dataflows.
<i>key</i>	(NULL character()) The series keys to query.
<i>start_period</i>	(NULL character(1) integer(1)) The start date of the data. Supported formats: <ul style="list-style-type: none"> • YYYY for annual data (e.g., ‘2019’) • YYYY-S[1-2] for semi-annual data (e.g., “2019-S1”) • YYYY-Q[1-4] for quarterly data (e.g., “2019-Q1”) • YYYY-MM for monthly data (e.g., “2019-01”) • YYYY-W[01-53] for weekly data (e.g., “2019-W01”) • YYYY-MM-DD for daily and business data (e.g., “2019-01-01”) If NULL, no start date restriction is applied (data retrieved from the earliest available date). Default NULL.
<i>end_period</i>	(NULL character(1) integer(1)) The end date of the data, in the same format as <i>start_period</i> . If NULL, no end date restriction is applied (data retrieved up to the most recent available date). Default NULL.
<i>first_n</i>	(NULL numeric(1)) Number of observations to retrieve from the start of the series. If NULL, no restriction is applied. Default NULL.

`last_n` (NULL | numeric(1))
Number of observations to retrieve from the end of the series. If NULL, no restriction is applied. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bundesbank.de/en/statistics/time-series-databases/help-for-sdmx-web-service/web-service-interface-data>

See Also

Other data: `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
# fetch all data for a given flow and key
data <- bbk_data("BBSIS", "D.I.ZAR.ZI.EUR.S1311.B.A604.R10XX.R.A.A._Z._Z.A")
head(data)
```

```
# fetch data for multiple keys
data <- bbk_data("BBEX3", c("M.ISK.EUR", "USD.CA.AC.A01"))
head(data)
```

```
# specified period (start date-end date) for daily data
data <- bbk_data(
  "BBSIS", "D.I.ZAR.ZI.EUR.S1311.B.A604.R10XX.R.A.A._Z._Z.A",
  start_period = "2020-01-01",
  end_period = "2020-08-01"
)
head(data)
```

```
# or only specify the start date
data <- bbk_data(
  "BBSIS", "D.I.ZAR.ZI.EUR.S1311.B.A604.R10XX.R.A.A._Z._Z.A",
  start_period = "2024-04-01"
)
head(data)
```

bbk_dimension	<i>Fetch Deutsche Bundesbank (BBk) dimensions</i>
---------------	---

Description

Retrieve the dimension structure for a given dataflow from the Bundesbank SDMX Web Service.

Usage

```
bbk_dimension(id)
```

Arguments

id	(character(1)) The id of the data structure definition to query (e.g., "BBK_BBSIS").
----	---

Value

A `data.table::data.table()` with columns:

id	The dimension id (e.g., "BBK_STD_FREQ", "BBK_STD_AREA")
position	The position of the dimension in the series key
codelist	The id of the associated codelist

Source

<https://www.bundesbank.de/en/statistics/time-series-databases/help-for-sdmx-web-service/web-service-interface-metadata>

See Also

Other metadata: `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
bbk_dimension("BBK_ERX")
```

bbk_metadata	<i>Fetch Deutsche Bundesbank (BBk) metadata</i>
--------------	---

Description

Retrieve metadata from the Bundesbank time series database via the SDMX Web Service.

Usage

```
bbk_metadata(type, id = NULL, lang = "en")
```

Arguments

type	(character(1)) The type of metadata to query. One of: "datastructure", "dataflow", "codelist", or "concept".
id	(NULL character(1)) The id to query. Default NULL.
lang	(character(1)) Language to query, either "en" or "de". Default "en".

Value

A `data.table::data.table()` with the requested metadata. The columns are:

id	The id of the metadata
name	The name of the metadata

Source

<https://www.bundesbank.de/en/statistics/time-series-databases/help-for-sdmx-web-service/web-service-interface-metadata>

See Also

Other metadata: [bbk_dimension\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_dimension\(\)](#), [nob_metadata\(\)](#), [onb_dimension\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [onb_toc\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#), [srb_series\(\)](#)

Examples

```
bbk_metadata("datastructure")
bbk_metadata("dataflow", "BBSIS")
bbk_metadata("codelist", "CL_BBK_ACIP_ASSET_LIABILITY")
bbk_metadata("concept", "CS_BBK_BSPL")
```

`bbk_series`*Fetch the Deutsche Bundesbank (BBk) series*

Description

Retrieve a single series by its key via the Bundesbank SDMX Web Service.

Usage

```
bbk_series(key)
```

Arguments

key	(NULL character()) The series keys to query.
-----	---

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bundesbank.de/en/statistics/time-series-databases/help-for-sdmx-web-service/web-service-interface-data>

See Also

`bbk_data()` for an endpoint with more options.

Other data: `bbk_data()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
bbk_series("BBEX3.M.DKK.EUR.BB.AC.A01")
bbk_series("BBAF3.Q.F41.S121.DE.S1.W0.LE.N._X.B")
bbk_series("BBBK11.D.TTA000")
```

bde_data	<i>Fetch Banco de España (BdE) data</i>
----------	---

Description

Retrieve time series data from the BdE statistics API.

Usage

```
bde_data(key, time_range = NULL, lang = "en")
```

Arguments

key	(character()) The series keys to query.
time_range	(NULL character(1) integer(1)) The time range for the data. Can be an annual range (e.g., 2024) or a frequency-based code: <ul style="list-style-type: none"> • Daily frequency (D): "3M" (last 3 months), "12M", "36M" • Monthly frequency (M): "30M", "60M", "MAX" (entire series) • Quarterly frequency (Q): "30M", "60M", "MAX" • Annual frequency (A): "60M", "MAX" If NULL (default), returns the smallest range for the series frequency (e.g., "30M" for monthly series).
lang	(character(1)) Language to query, either "en" or "es".

Details

You can search for the series codes in the **BIEST** application or in the tables published by the Banco de España.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bde.es/webbe/en/estadisticas/recursos/api-estadisticas-bde.html>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
bde_data("D_1NBAF472", time_range = "30M")
bde_data(c("DTNPDE2010_P0000P_PS_APU", "DTNSEC2010_S0000P_APU_SUMAMOVIL"), time_range = "MAX")
bde_data("DEEQ.N.ES.W1.S1.S1.T.B.G._Z._Z._Z.EUR._T._X.N.ALL", time_range = 2024)
```

bde_latest

Fetch latest Banco de España (BdE) data

Description

Retrieve the most recently published value for one or more series from the BdE statistics API.

Usage

```
bde_latest(key, lang = "en")
```

Arguments

key	(character()) The series keys to query.
lang	(character(1)) Language to query, either "en" or "es".

Value

A `data.table::data.table()` with the latest observation per series.

Source

<https://www.bde.es/webbe/en/estadisticas/recursos/api-estadisticas-bde.html>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
bde_latest("D_1NBAF472")
bde_latest(c("D_1NBAF472", "DTNPDE2010_P0000P_PS_APU"))
```

bdf_codelist	<i>Fetch Banque de France (BdF) codelists</i>
--------------	---

Description

Fetch Banque de France (BdF) codelists

Usage

```
bdf_codelist(..., lang = "en")
```

Arguments

...	(any) Extra arguments appended to the API request. Combined with the default arguments with <code>modifyList()</code> .
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

Source

<https://webstat.banque-france.fr/en/pages/guide-migration-api/>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
## Not run:  
bdf_codelist()  
  
# filter for a specific codelist  
bdf_codelist(where = "codelist_id = 'CL_FREQ'")  
  
## End(Not run)
```

bdf_data	<i>Fetch Banque de France (BdF) data</i>
----------	--

Description

Retrieve time series data from the BdF Webstat API.

Usage

```
bdf_data(  
  ...,  
  key = NULL,  
  start_date = NULL,  
  end_date = NULL,  
  lang = "en",  
  api_key = bdf_key()  
)
```

Arguments

...	(any) Extra arguments appended to the API request. Combined with the default arguments with <code>modifyList()</code> .
key	(NULL character(1)) The series key to query. Default NULL.
start_date	(NULL character(1) Date(1)) Start date of the data. Default NULL.
end_date	(NULL character(1) Date(1)) End date of the data. Default NULL.
lang	(character(1)) Language to query. Default "en".
api_key	(character(1)) API key to use for the request. Defaults to the value returned by <code>bdf_key()</code> , which reads from the <code>BANQUEDEFrance_KEY</code> environment variable.

Value

A `data.table::data.table()` with the requested data.

Source

<https://webstat.banque-france.fr/en/pages/guide-migration-api/>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_dataset\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [boe_data\(\)](#), [ecb_data\(\)](#), [nob_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_cross_rates\(\)](#), [srb_data\(\)](#)

Examples

```
## Not run:
bdf_data(key = "CONJ2.M.R24.T.SM.0RG24.EFTPM100.10")

# inflation rate
bdf_data(key = "ICP.M.FR.N.000000.4.ANR")

# or with a date filter
bdf_data(key = "ICP.M.FR.N.000000.4.ANR", start_date = "2025-01-01", end_date = "2025-06-30")

# advanced filter with where clause
bdf_data(key = "ICP.M.FR.N.000000.4.ANR", where = "time_period_start >= date'2025-01-01'")

## End(Not run)
```

bdf_dataset

Fetch Banque de France (BdF) datasets

Description

Fetch Banque de France (BdF) datasets

Usage

```
bdf_dataset(..., lang = "en")
```

Arguments

...	(any) Extra arguments appended to the API request. Combined with the default arguments with modifyList() .
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

Source

<https://webstat.banque-france.fr/en/pages/guide-migration-api/>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_data\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [boe_data\(\)](#), [ecb_data\(\)](#), [nob_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_cross_rates\(\)](#), [srb_data\(\)](#)

Examples

```
## Not run:
bdf_dataset()

# structure of a dataset
bdf_dataset(where = "dataset_id = 'CONJ2'")

## End(Not run)
```

bis_data

*Fetch Bank for International Settlements (BIS) data***Description**

Retrieve time series data from the BIS SDMX Web Service.

Usage

```
bis_data(
  flow,
  key = NULL,
  start_period = NULL,
  end_period = NULL,
  first_n = NULL,
  last_n = NULL
)
```

Arguments

flow	(character(1)) The dataflow to query. See bis_metadata() for available dataflows.
key	(NULL character()) The series keys to query using dot-separated dimension values (e.g., "M.CH"). Use + for multiple values in one dimension (e.g., "M.CH+US"). If NULL, all data for the flow is returned. Default NULL.
start_period	(NULL character(1) integer(1)) Start date of the data. Supported formats: <ul style="list-style-type: none"> • YYYY for annual data (e.g., 2019) • YYYY-S[1-2] for semi-annual data (e.g., "2019-S1") • YYYY-Q[1-4] for quarterly data (e.g., "2019-Q1")

- YYYY-MM for monthly data (e.g., "2019-01")
- YYYY-MM-DD for daily data (e.g., "2019-01-01")

If NULL, no start date restriction is applied (data retrieved from the earliest available date). Default NULL.

end_period	(NULL character(1) integer(1)) End date of the data, in the same format as start_period. If NULL, no end date restriction is applied (data retrieved up to the most recent available date). Default NULL.
first_n	(NULL numeric(1)) Number of observations to retrieve from the start of the series. If NULL, no restriction is applied. Default NULL.
last_n	(NULL numeric(1)) Number of observations to retrieve from the end of the series. If NULL, no restriction is applied. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://stats.bis.org/api-doc/v1/>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
# fetch Swiss central bank policy rate
bis_data("WS_CBPOL", "M.CH", last_n = 5L)

# fetch effective exchange rates
bis_data("WS_EER", "M.N.B.CH", start_period = "2020-01")
```

bis_dimension

Fetch Bank for International Settlements (BIS) dimensions

Description

Retrieve the dimension structure for a given dataflow from the BIS SDMX Web Service.

Usage

```
bis_dimension(id)
```

Arguments

`id` (character(1))
The id of the data structure definition to query (e.g., "BIS_CBPOL").

Value

A `data.table::data.table()` with columns:

<code>id</code>	The dimension id (e.g., "FREQ", "REF_AREA")
<code>position</code>	The position of the dimension in the series key
<code>codelist</code>	The id of the associated codelist (e.g., "CL_FREQ")

Source

<https://stats.bis.org/api-doc/v1/>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
bis_dimension("BIS_CBPOL")
```

bis_metadata

Fetch Bank for International Settlements (BIS) metadata

Description

Retrieve metadata from the BIS SDMX Web Service.

Usage

```
bis_metadata(type, id = NULL)
```

Arguments

type	(character(1)) The type of metadata to query. One of: "datastructure", "dataflow", "codelist", or "concept".
id	(NULL character(1)) The id to query. Default NULL.

Value

A `data.table::data.table()` with the requested metadata.

Source

<https://stats.bis.org/api-doc/v1/>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
bis_metadata("dataflow")
bis_metadata("datastructure", "BIS_CBPOL")
bis_metadata("codelist", "CL_FREQ")
```

boc_catalog

Fetch Bank of Canada (BoC) available series or group

Description

Access all available series or groups from the Bank of Canada Valet API.

Usage

```
boc_catalog(type = "groups")
```

Arguments

type	(character(1)) Set of data to return. One of "groups" or "series". Default "groups".
------	---

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bankofcanada.ca/valet/docs>

Examples

```
## Not run:
catalog <- boc_catalog()
head(catalog)

# filter for effective exchange rate series
dt <- catalog[grepl("CEER", label)]
head(dt)

## End(Not run)
```

boc_data	<i>Fetch Bank of Canada (BoC) data</i>
----------	--

Description

Retrieve time series data from the Bank of Canada Valet API.

Usage

```
boc_data(
  group_name = NULL,
  series_name = NULL,
  start_date = NULL,
  end_date = NULL
)
```

Arguments

- group_name (NULL | character(1))
Name of the group. Only one of group_name or series_name can be used.
- series_name (NULL | character())
Name of the series.
- start_date (NULL | Date(1) | character(1))
Start date of the data. Default NULL.
- end_date (NULL | Date(1) | character(1))
End date of the data. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bankofcanada.ca/valet/docs>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
## Not run:
# fetch all data for a single group
dt <- boc_data(group_name = "FX_RATES_DAILY")
head(dt)

# or for multiple series ids
dt <- boc_data(
  series_name = c("FXUSDCAD", "FXEURCAD"),
  start_date = "2023-01-23",
  end_date = "2023-07-19"
)
head(dt)

## End(Not run)
```

boc_fx_rates

Fetch Bank of Canada foreign exchange rates

Description

Fetch the latest or historical foreign exchange reference rates from the Bank of Canada (BoC).

Usage

```
boc_fx_rates(start_date = NULL, end_date = NULL, limit = NULL, skip = NULL)
```

Arguments

<code>start_date</code>	(NULL Date(1) character(1)) Start date of the data. Default NULL.
<code>end_date</code>	(NULL Date(1) character(1)) End date of the data. Default NULL.

limit (NULL | integer(1))
 Maximum number of records to return. Default NULL (all records).

skip (NULL | integer(1))
 Number of records to skip. Default NULL (do not skip any records).

Details

The recorded rates indicate the number of Canadian dollars required to buy a single unit of the foreign currency. New rates are released by the Bank of Canada (BoC) daily at 4:30 pm. The Canada Border Services (CBSA) retrieves these updates between 4:30 pm and 5 pm ET.

BoC provides 23 foreign exchange rates. All other rates are maintained by the CBSA.

Exchange rates from the BoC are updated daily in the system while other exchange rates are updated by the CBSA at set intervals. The updated rates are available for retrieval between 7 pm and 11:59 pm ET.

As BoC publishes exchange rates every business day, it is recommended that exchange rate data be retrieved on a daily basis. This retrieval should occur after 7 pm ET to ensure retrieval of the latest updates.

Value

A `data.table::data.table()` with the exchange rates.

Source

<https://www.cbsa-asfc.gc.ca/eservices/api/er-tc-api-eng.html>

Examples

```
# fetch latest exchange rates
boc_fx_rates()

# fetch historical exchange rates
boc_fx_rates(start_date = "2021-10-22", end_date = "2021-10-23", limit = 10, skip = 2)
```

boc_metadata *Fetch Bank of Canada (BoC) metadata (details)*

Description

Fetch Bank of Canada (BoC) metadata (details)

Usage

```
boc_metadata(group_name = NULL, series_name = NULL)
```

Arguments

group_name (NULL | character(1))
Name of the group. Only one of group_name or series_name can be used.

series_name (NULL | character())
Name of the series.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bankofcanada.ca/valet/docs>

Examples

```
## Not run:
  boc_metadata(group_name = "FX_RATES_DAILY")
  boc_metadata(series_name = "FXUSDCAD")

## End(Not run)
```

boe_data	<i>Fetch Bank of England (BoE) data</i>
----------	---

Description

Retrieve time series data from the BoE database.

Usage

```
boe_data(key, start_date, end_date = Sys.Date())
```

Arguments

key (character())
The series keys to query.

start_date (character(1) | Date(1))
Start date of the data.

end_date (character(1) | Date(1))
End date of the data. Default is today's date.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.bankofengland.co.uk/boeapps/database>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_data\(\)](#), [bdf_dataset\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [ecb_data\(\)](#), [nob_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_cross_rates\(\)](#), [srb_data\(\)](#)

Examples

```
# Bank Rate
boe_data("IUBNDR", "2015-01-01")

# SONIA daily rate
boe_data("IUDSOIA", "2015-01-01")

# 10-year nominal par yield
boe_data("IUDMNPY", "2015-01-01")

# multiple series
boe_data(c("IUMABEDR", "IUALBEDR"), "2015-01-01")
```

cache

Get or manage the bbk API cache

Description

`bbk_cache_dir()` returns the path where cached API responses are stored. `bbk_cache_clear()` clears all cached responses.

Usage

```
bbk_cache_dir()

bbk_cache_clear()
```

Details

The cache is only used when enabled with `options(bbk.cache = TRUE)`. Cached responses are stored for 1 day by default, but this can be customized with `options(bbk.cache_max_age = seconds)`.

Examples

```
## Not run:
# enable caching
options(bbk.cache = TRUE)

# view cache location
bbk_cache_dir()

# clear the cache
bbk_cache_clear()

## End(Not run)
```

`ecb_data`*Fetch European Central Bank (ECB) data*

Description

Retrieve time series data from the ECB SDMX Web Service.

Usage

```
ecb_data(
  flow,
  key = NULL,
  start_period = NULL,
  end_period = NULL,
  first_n = NULL,
  last_n = NULL,
  updated_after = NULL
)
```

Arguments

<code>flow</code>	(character(1)) Flow to query.
<code>key</code>	(NULL character()) The series keys to query.
<code>start_period</code>	(NULL character(1) integer(1)) Start date of the data. Supported formats: <ul style="list-style-type: none"> • YYYY for annual data (e.g., 2019) • YYYY-S[1-2] for semi-annual data (e.g., "2019-S1") • YYYY-Q[1-4] for quarterly data (e.g., "2019-Q1") • YYYY-MM for monthly data (e.g., "2019-01") • YYYY-W[01-53] for weekly data (e.g., "2019-W01") • YYYY-MM-DD for daily and business data (e.g., "2019-01-01")

	If NULL, no start date restriction is applied (data retrieved from the earliest available date). Default NULL.
end_period	(NULL character(1) integer(1)) End date of the data, in the same format as start_period. If NULL, no end date restriction is applied (data retrieved up to the most recent available date). Default NULL.
first_n	(NULL numeric(1)) Number of observations to retrieve from the start of the series. If NULL, no restriction is applied. Default NULL.
last_n	(NULL numeric(1)) Number of observations to retrieve from the end of the series. If NULL, no restriction is applied. Default NULL.
updated_after	(NULL character(1)) ISO 8601 timestamp to retrieve only observations updated after the given time (e.g., "2024-06-01T00:00:00"). If NULL, no restriction is applied. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://data.ecb.europa.eu/help/api/data>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `nob_data()`, `onb_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
# fetch US dollar/Euro exchange rate
ecb_data("EXR", "D.USD.EUR.SP00.A")

# fetch data for multiple keys
ecb_data("EXR", c("D.USD", "JPY.EUR.SP00.A"))
```

ecb_dimension	<i>Fetch European Central Bank (ECB) dimensions</i>
---------------	---

Description

Retrieve the dimension structure for a given dataflow from the ECB SDMX Web Service.

Usage

```
ecb_dimension(id)
```

Arguments

id	(character(1)) The id of the data structure definition to query (e.g., "ECB_EXR1").
----	--

Value

A `data.table::data.table()` with columns:

id	The dimension id (e.g., "FREQ", "CURRENCY")
position	The position of the dimension in the series key
codelist	The id of the associated codelist (e.g., "CL_FREQ")

Source

<https://data.ecb.europa.eu/help/api/metadata>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
ecb_dimension("ECB_EXR1")
```

ecb_fx_rates	<i>Fetch Euro foreign exchange reference rates</i>
--------------	--

Description

Fetch the latest or historical Euro foreign exchange reference rates from the European Central Bank (ECB).

Usage

```
ecb_fx_rates(x = "latest")
```

```
ecb_euro_rates(x = "latest")
```

Arguments

x	(character(1)) One of "latest" or "history". Default "latest".
---	---

Details

Note you can achieve the same by calling the `ecb_data()` function with the right parameters for each currency.

The reference rates are usually updated at around 16:00 CET every working day, except on **TARGET closing days**.

They are based on the daily concertation procedure between central banks across Europe, which normally takes place around 14:10 CET. The reference rates are published for information purposes only. Using the rates for transaction purposes is strongly discouraged.

Value

A `data.table::data.table()` with the exchange rates.

Source

https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html

Examples

```
ecb_fx_rates()
```

ecb_metadata	<i>Fetch European Central Bank (ECB) metadata</i>
--------------	---

Description

Retrieve metadata from the ECB time series database via the SDMX Web Service.

Usage

```
ecb_metadata(type, agency = NULL, id = NULL)
```

Arguments

type	(character(1)) The type of metadata to query. One of: "datastructure", "dataflow", "codelist", or "concept".
agency	(NULL character(1)) The id of the agency to query. Default NULL.
id	(NULL character(1)) The id of the resource to query. Default NULL.

Value

A `data.table::data.table()` with the requested metadata. The columns are:

agency	The agency of the metadata
id	The id of the metadata
name	The name of the metadata

Source

<https://data.ecb.europa.eu/help/api/metadata>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
ecb_metadata("datastructure")
ecb_metadata("datastructure", "ECB")
ecb_metadata("datastructure", "ECB", "ECB_EXR1")
ecb_metadata("datastructure", id = "ECB_EXR1")
```

nob_data	<i>Fetch Norges Bank (NoB) data</i>
----------	-------------------------------------

Description

Retrieve time series data from the Norges Bank SDMX Web Service.

Usage

```
nob_data(
  flow,
  key = NULL,
  start_period = NULL,
  end_period = NULL,
  first_n = NULL,
  last_n = NULL
)
```

Arguments

flow	(character(1)) The dataflow to query. See nob_metadata() for available dataflows.
key	(NULL character(1)) The series key to query using dot-separated dimension values (e.g., "B.USD.NOK.SP"). Use + for multiple values in one dimension (e.g., "B.USD+EUR.NOK.SP"). If NULL, all data for the flow is returned. Default NULL.
start_period	(NULL character(1) integer(1)) Start date of the data (e.g., "2024-01-01" or 2024). If NULL, no start date restriction is applied. Default NULL.
end_period	(NULL character(1) integer(1)) End date of the data, in the same format as start_period. If NULL, no end date restriction is applied. Default NULL.
first_n	(NULL numeric(1)) Number of observations to retrieve from the start of the series. If NULL, no restriction is applied. Default NULL.
last_n	(NULL numeric(1)) Number of observations to retrieve from the end of the series. If NULL, no restriction is applied. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.norges-bank.no/en/topics/Statistics/open-data/>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_data\(\)](#), [bdf_dataset\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [boe_data\(\)](#), [ecb_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_cross_rates\(\)](#), [srb_data\(\)](#)

Examples

```
# fetch USD/NOK exchange rate
nob_data("EXR", "B.USD.NOK.SP", last_n = 5L)

# fetch multiple exchange rates
nob_data("EXR", "B.USD+EUR+GBP.NOK.SP", start_period = "2024-01-01")

# fetch policy rate
nob_data("IR", last_n = 5L)
```

nob_dimension

Fetch Norges Bank (NoB) dimensions

Description

Retrieve the dimension structure for a given dataflow from the Norges Bank SDMX Web Service.

Usage

```
nob_dimension(id)
```

Arguments

id	(character(1)) The id of the data structure definition to query (e.g., "NB_EXR").
----	--

Value

A `data.table::data.table()` with columns:

id	The dimension id (e.g., "FREQ", "BASE_CUR")
position	The position of the dimension in the series key
codelist	The id of the associated codelist (e.g., "CL_FREQ")

Source

<https://www.norges-bank.no/en/topics/Statistics/open-data/>

See Also

Other metadata: [bbk_dimension\(\)](#), [bbk_metadata\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_metadata\(\)](#), [onb_dimension\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [onb_toc\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#), [srb_series\(\)](#)

Examples

```
nob_dimension("DSD_EXR")
```

nob_metadata	<i>Fetch Norges Bank (NoB) metadata</i>
--------------	---

Description

Retrieve metadata from the Norges Bank SDMX Web Service.

Usage

```
nob_metadata(type, id = NULL, lang = "en")
```

Arguments

type	(character(1)) The type of metadata to query. One of: "datastructure", "dataflow", "codelist", or "concept".
id	(NULL character(1)) The id to query. Default NULL.
lang	(character(1)) Language for names, either "en" or "no". Default "en".

Value

A `data.table::data.table()` with the requested metadata.

Source

<https://www.norges-bank.no/en/topics/Statistics/open-data/>

See Also

Other metadata: [bbk_dimension\(\)](#), [bbk_metadata\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_dimension\(\)](#), [onb_dimension\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [onb_toc\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#), [srb_series\(\)](#)

Examples

```
nob_metadata("dataflow")
nob_metadata("datastructure")
nob_metadata("codelist", "CL_CURRENCY")
```

onb_data

Fetch Österreichische Nationalbank (OeNB) data

Description

Retrieve time series data from the OeNB Web Service.

Usage

```
onb_data(
  hier_id,
  key,
  ...,
  start_period = NULL,
  end_period = NULL,
  freq = NULL,
  lang = "en"
)
```

Arguments

hier_id	(integer(1)) Hierarchy id to query.
key	(character()) The series keys to query.
...	(any) Additional parameters to pass to the API.
start_period	(NULL character(1) integer(1)) Start date of the data.
end_period	(NULL character(1) integer(1)) End date of the data.
freq	(NULL character(1)) Frequency of the data.
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

Source

<https://www.oenb.at/en/Statistics/User-Defined-Tables/webservice.html>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `snb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
onb_data(hier_id = 11, key = "VDBFKBSC217000")

# Loans to euro area residents, since 2000:
onb_data(hier_id = 11, key = "VDBFKBSC217000", start_period = "2000-01-01")

# Austrian imports and exports of goods from/to Germany, 2002-2012, annual frequency:
onb_data(hier_id = 901, key = "VDBQZA1000", start_period = 2002, end_period = 2012, freq = "A")

# Number of Austrian banks' subsidiaries abroad and in the EU, from 2005, semiannual:
onb_data(
  hier_id = 321,
  key = c("VDBKISDANZTAU", "VDBKISDANZTEU"),
  start_period = 200501,
  freq = "H"
)
```

onb_dimension

Fetch Österreichische Nationalbank (OeNB) dimension

Description

Fetch Österreichische Nationalbank (OeNB) dimension

Usage

```
onb_dimension(hier_id, key, lang = "en")
```

Arguments

hier_id	(integer(1)) Hierarchy id to query.
key	(character()) The series keys to query.
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

See Also

Other metadata: [bbk_dimension\(\)](#), [bbk_metadata\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_dimension\(\)](#), [nob_metadata\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [onb_toc\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#), [srb_series\(\)](#)

Examples

```
onb_dimension(hier_id = 11, key = "VDBFKBSC217000")
```

onb_frequency	<i>Fetch Österreichische Nationalbank (OeNB) data frequency</i>
---------------	---

Description

Fetch Österreichische Nationalbank (OeNB) data frequency

Usage

```
onb_frequency(hier_id, key, ..., lang = "en")
```

Arguments

hier_id	(integer(1)) Hierarchy id to query.
key	(character()) The series keys to query.
...	(any) Additional parameters to pass to the API.
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
onb_frequency(hier_id = 74, key = "VDBOSBHAGBSTIN")
onb_frequency(hier_id = 11, key = "VDBFKBSC217000")
```

onb_hierarchy	<i>Fetch Österreichische Nationalbank (OeNB) hierarchy</i>
---------------	--

Description

Fetch Österreichische Nationalbank (OeNB) hierarchy

Usage

```
onb_hierarchy(hier_id, lang = "en")
```

Arguments

hier_id	(integer(1)) Hierarchy id to query.
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
onb_hierarchy(hier_id = 11)
```

onb_metadata	<i>Fetch Österreichische Nationalbank (OeNB) metadata</i>
--------------	---

Description

Fetch Österreichische Nationalbank (OeNB) metadata

Usage

```
onb_metadata(hier_id, key, ..., lang = "en")
```

Arguments

hier_id	(integer(1)) Hierarchy id to query.
key	(character()) The series keys to query.
...	(any) Additional parameters to pass to the API.
lang	(character(1)) Language to query. Default "en".

Value

A `data.table::data.table()` with the requested data.

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_toc()`, `snb_dimension()`, `srb_calendar()`, `srb_series()`

Examples

```
onb_metadata(hier_id = 11, key = "VDBFKBSC217000")
```

onb_toc	<i>Fetch Österreichische Nationalbank (OeNB) table of contents</i>
---------	--

Description

Fetch Österreichische Nationalbank (OeNB) table of contents

Usage

```
onb_toc(lang = "en")
```

Arguments

lang	(character(1)) Language to query. Default "en".
------	--

Value

A `data.table::data.table()` with the requested data.

See Also

Other metadata: [bbk_dimension\(\)](#), [bbk_metadata\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_dimension\(\)](#), [nob_metadata\(\)](#), [onb_dimension\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#), [srb_series\(\)](#)

Examples

```
onb_toc()
```

snb_data	<i>Fetch Swiss National Bank (SNB) data</i>
----------	---

Description

Retrieve time series data from the SNB data portal.

Usage

```
snb_data(key, start_date = NULL, end_date = NULL, lang = "en")
```

Arguments

key	(character(1)) The series key to query.
start_date	(NULL character(1) Date(1)) Start date of the data.
end_date	(NULL character(1) Date(1)) End date of the data.
lang	(character(1)) Language to query, either "en" or "de". Default "en".

Value

A `data.table::data.table()` with the requested data.

Source

<https://data.snb.ch/en>

See Also

Other data: `bbk_data()`, `bbk_series()`, `bde_data()`, `bde_latest()`, `bdf_codelist()`, `bdf_data()`, `bdf_dataset()`, `bis_data()`, `boc_data()`, `boe_data()`, `ecb_data()`, `nob_data()`, `onb_data()`, `srb_cross_rates()`, `srb_data()`

Examples

```
snb_data("rendopar")

# or filter for date range
snb_data("rendopar", "2020-01-01", "2020-12-31")
```

snb_dimension

Fetch Swiss National Bank (SNB) dimensions

Description

Retrieve the dimension structure for a given cube from the SNB data portal.

Usage

```
snb_dimension(key, lang = "en")
```

Arguments

key	(character(1)) The series key to query.
lang	(character(1)) Language to query, either "en" or "de". Default "en".

Value

A `data.table::data.table()` with the dimension structure.

Source

<https://data.snb.ch/en>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `srb_calendar()`, `srb_series()`

Examples

```
snb_dimension("rendopar")
```

srb_calendar	<i>Fetch Sveriges Riksbank (SRB) calendar days</i>
--------------	--

Description

Retrieve Swedish banking calendar information from the Sveriges Riksbank SWEA API.

Usage

```
srb_calendar(start_date, end_date = NULL)
```

Arguments

start_date	(Date(1) character(1)) Start date of the range (e.g., "2024-01-01").
end_date	(NULL Date(1) character(1)) End date of the range. If NULL, data up to the latest available date is returned. Default NULL.

Value

A `data.table::data.table()` with the calendar day information.

Source

<https://developer.api.riksbank.se/>

See Also

Other metadata: `bbk_dimension()`, `bbk_metadata()`, `bis_dimension()`, `bis_metadata()`, `ecb_dimension()`, `ecb_metadata()`, `nob_dimension()`, `nob_metadata()`, `onb_dimension()`, `onb_frequency()`, `onb_hierarchy()`, `onb_metadata()`, `onb_toc()`, `snb_dimension()`, `srb_series()`

Examples

```
srb_calendar("2024-01-01", "2024-01-31")
```

srb_cross_rates	<i>Fetch Sveriges Riksbank (SRB) cross rates</i>
-----------------	--

Description

Compute cross exchange rates between two currency series from the Sveriges Riksbank SWEA API.

Usage

```
srb_cross_rates(series1, series2, start_date, end_date = NULL)
```

Arguments

series1	(character(1)) The first series ID (e.g., "SEKUSDPMI").
series2	(character(1)) The second series ID (e.g., "SEKEURPMI").
start_date	(NULL Date(1) character(1)) Start date of the data (e.g., "2024-01-01"). If NULL, all available data is returned. Default NULL.
end_date	(NULL Date(1) character(1)) End date of the data, in the same format as start_date. If NULL, data up to the latest available date is returned. Default NULL.

Value

A `data.table::data.table()` with the cross rate data.

Source

<https://developer.api.riksbank.se/>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_data\(\)](#), [bdf_dataset\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [boe_data\(\)](#), [ecb_data\(\)](#), [nob_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_data\(\)](#)

Examples

```
# USD/EUR cross rate
srb_cross_rates("SEKUSDPMI", "SEKEURPMI", start_date = "2024-01-01", end_date = "2024-01-31")
```

srb_data

Fetch Sveriges Riksbank (SRb) data

Description

Retrieve time series data from the Sveriges Riksbank SWEA API.

Usage

```
srb_data(series, start_date = NULL, end_date = NULL)
```

Arguments

series	(character(1)) The series ID to query. See srb_series() for available series.
start_date	(NULL Date(1) character(1)) Start date of the data (e.g., "2024-01-01"). If NULL, all available data is returned. Default NULL.
end_date	(NULL Date(1) character(1)) End date of the data, in the same format as start_date. If NULL, data up to the latest available date is returned. Default NULL.

Value

A `data.table::data.table()` with the requested data.

Source

<https://developer.api.riksbank.se/>

See Also

Other data: [bbk_data\(\)](#), [bbk_series\(\)](#), [bde_data\(\)](#), [bde_latest\(\)](#), [bdf_codelist\(\)](#), [bdf_data\(\)](#), [bdf_dataset\(\)](#), [bis_data\(\)](#), [boc_data\(\)](#), [boe_data\(\)](#), [ecb_data\(\)](#), [nob_data\(\)](#), [onb_data\(\)](#), [snb_data\(\)](#), [srb_cross_rates\(\)](#)

Examples

```
# fetch USD/SEK exchange rate
srb_data("SEKUSDPMI", start_date = "2024-01-01")

# fetch EUR/SEK exchange rate
srb_data("SEKEURPMI", start_date = "2024-01-01")
```

srb_series

Fetch Sveriges Riksbank (SRb) series metadata

Description

Retrieve available series or group metadata from the Sveriges Riksbank SWEA API.

Usage

```
srb_series(type = "series")
```

Arguments

type (character(1))
The type of metadata to query. One of "series" or "groups". Default "series".

Value

A `data.table::data.table()` with the requested metadata.

Source

<https://developer.api.riksbank.se/>

See Also

Other metadata: [bbk_dimension\(\)](#), [bbk_metadata\(\)](#), [bis_dimension\(\)](#), [bis_metadata\(\)](#), [ecb_dimension\(\)](#), [ecb_metadata\(\)](#), [nob_dimension\(\)](#), [nob_metadata\(\)](#), [onb_dimension\(\)](#), [onb_frequency\(\)](#), [onb_hierarchy\(\)](#), [onb_metadata\(\)](#), [onb_toc\(\)](#), [snb_dimension\(\)](#), [srb_calendar\(\)](#)

Examples

```
srb_series()
```

Index

* data

- bbk_data, 3
- bbk_series, 7
- bde_data, 8
- bde_latest, 9
- bdf_codelist, 10
- bdf_data, 11
- bdf_dataset, 12
- bis_data, 13
- boc_data, 17
- boe_data, 20
- ecb_data, 22
- nob_data, 27
- onb_data, 30
- snb_data, 35
- srb_cross_rates, 38
- srb_data, 39

* metadata

- bbk_dimension, 5
- bbk_metadata, 6
- bis_dimension, 14
- bis_metadata, 15
- ecb_dimension, 24
- ecb_metadata, 26
- nob_dimension, 28
- nob_metadata, 29
- onb_dimension, 31
- onb_frequency, 32
- onb_hierarchy, 33
- onb_metadata, 34
- onb_toc, 35
- snb_dimension, 36
- srb_calendar, 37
- srb_series, 40

- bbk_cache_clear (cache), 21
- bbk_cache_dir (cache), 21
- bbk_data, 3, 7–10, 12–14, 18, 21, 23, 28, 31, 36, 39, 40
- bbk_data(), 7

- bbk_dimension, 5, 6, 15, 16, 24, 26, 29, 32–35, 37, 38, 40
- bbk_metadata, 5, 6, 15, 16, 24, 26, 29, 32–35, 37, 38, 40
- bbk_metadata(), 3
- bbk_series, 4, 7, 8–10, 12–14, 18, 21, 23, 28, 31, 36, 39, 40
- bde_data, 4, 7, 8, 9, 10, 12–14, 18, 21, 23, 28, 31, 36, 39, 40
- bde_latest, 4, 7, 8, 9, 10, 12–14, 18, 21, 23, 28, 31, 36, 39, 40
- bdf_codelist, 4, 7–9, 10, 12–14, 18, 21, 23, 28, 31, 36, 39, 40
- bdf_data, 4, 7–10, 11, 13, 14, 18, 21, 23, 28, 31, 36, 39, 40
- bdf_dataset, 4, 7–10, 12, 12, 14, 18, 21, 23, 28, 31, 36, 39, 40
- bis_data, 4, 7–10, 12, 13, 13, 18, 21, 23, 28, 31, 36, 39, 40
- bis_dimension, 5, 6, 14, 16, 24, 26, 29, 32–35, 37, 38, 40
- bis_metadata, 5, 6, 15, 15, 24, 26, 29, 32–35, 37, 38, 40
- bis_metadata(), 13
- boc_catalog, 16
- boc_data, 4, 7–10, 12–14, 17, 21, 23, 28, 31, 36, 39, 40
- boc_fx_rates, 18
- boc_metadata, 19
- boe_data, 4, 7–10, 12–14, 18, 20, 23, 28, 31, 36, 39, 40
- cache, 21
- data.table::data.table(), 4–12, 14–20, 23–29, 31–40
- ecb_data, 4, 7–10, 12–14, 18, 21, 22, 28, 31, 36, 39, 40
- ecb_data(), 25

ecb_dimension, 5, 6, 15, 16, 24, 26, 29,
32–35, 37, 38, 40
ecb_euro_rates (ecb_fx_rates), 25
ecb_fx_rates, 25
ecb_metadata, 5, 6, 15, 16, 24, 26, 29, 32–35,
37, 38, 40

modifyList(), 10–12

nob_data, 4, 7–10, 12–14, 18, 21, 23, 27, 31,
36, 39, 40
nob_dimension, 5, 6, 15, 16, 24, 26, 28, 29,
32–35, 37, 38, 40
nob_metadata, 5, 6, 15, 16, 24, 26, 29, 29,
32–35, 37, 38, 40
nob_metadata(), 27

onb_data, 4, 7–10, 12–14, 18, 21, 23, 28, 30,
36, 39, 40
onb_dimension, 5, 6, 15, 16, 24, 26, 29, 31,
33–35, 37, 38, 40
onb_frequency, 5, 6, 15, 16, 24, 26, 29, 32,
32, 33–35, 37, 38, 40
onb_hierarchy, 5, 6, 15, 16, 24, 26, 29, 32,
33, 33, 34, 35, 37, 38, 40
onb_metadata, 5, 6, 15, 16, 24, 26, 29, 32, 33,
34, 35, 37, 38, 40
onb_toc, 5, 6, 15, 16, 24, 26, 29, 32–34, 35,
37, 38, 40

snb_data, 4, 7–10, 12–14, 18, 21, 23, 28, 31,
35, 39, 40
snb_dimension, 5, 6, 15, 16, 24, 26, 29,
32–35, 36, 38, 40
srb_calendar, 5, 6, 15, 16, 24, 26, 29, 32–35,
37, 37, 40
srb_cross_rates, 4, 7–10, 12–14, 18, 21, 23,
28, 31, 36, 38, 40
srb_data, 4, 7–10, 12–14, 18, 21, 23, 28, 31,
36, 39, 39
srb_series, 5, 6, 15, 16, 24, 26, 29, 32–35,
37, 38, 40
srb_series(), 39