

Package ‘GetBCBData’

October 5, 2026

Type Package

Title Imports Datasets from BCB (Central Bank of Brazil) using Its Official API

Version 0.9.5

Maintainer Marcelo Perlin <marceloperlin@gmail.com>

Description Downloads and organizes datasets using BCB's API <<https://www.bcb.gov.br/>>. Offers options for caching with the 'memoise' package and , multicore/multisession with 'furry' and format of output data (long/wide).

Depends R (>= 3.3.0)

Imports curl, dplyr, future, furry, jsonlite, memoise, purrr, cli, parallelly, xml2

License GPL-2

BugReports <https://github.com/msperlin/GetBCBData/issues>

URL <https://github.com/msperlin/GetBCBData/>

RoxygenNote 7.3.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), ggplot2

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Author Marcelo Perlin [aut, cre]

Repository CRAN

Date/Publication 2026-10-05 10:30:02 UTC

Contents

gcbbd_get_series	2
gcbbd_search_series	3
Index	5

gcbcb_get_series	<i>Imports time series data from BCB-SGS System (Banco Central do Brasil, sistema de series temporais)</i>
------------------	--

Description

Using BCB's official API at <<https://www.bcb.gov.br/>>, this function will download data for a specific set of ids and dates. The main advantage is the use of caching and parallel computing for fast operations. You can search for available series at <<https://www.bcb.gov.br/?sgs>>

Usage

```
gcbcb_get_series(
  id,
  first_date = Sys.Date() - 5 * 365,
  last_date = Sys.Date(),
  format_data = "long",
  be_quiet = FALSE,
  use_memoise = TRUE,
  cache_path = gcbcb_get_default_cache_folder(),
  do_parallel = FALSE,
  first.date = NULL,
  last.date = NULL,
  format.data = NULL,
  be.quiet = NULL,
  use.memoise = NULL,
  cache.path = NULL,
  do.parallel = NULL
)
```

Arguments

id	Id of time series. The name of the vector sets the name of the series in the output (e.g <code>i_d <- c('SELIC' = 11)</code>). You can search for ids in the official BCB-SGS webpage < https://www.bcb.gov.br/?sgs >
first_date	First date of time series
last_date	Last date of time series
format_data	The format of the datasets - long (default, series incremented by rows) or wide (series incremented by columns)
be_quiet	Logical. Should functions output messages to screen? - FALSE (default) or TRUE
use_memoise	Logical. Sets the use of caching system - TRUE (default) or FALSE
cache_path	Path to save cache files - 'gcbcb_cache' (default)
do_parallel	Logical for parallel data importation - FALSE (default)
first.date	Deprecated. Use first_date instead.

last.date	Deprecated. Use last_date instead.
format.data	Deprecated. Use format_data instead.
be.quiet	Deprecated. Use be_quiet instead.
use.memoise	Deprecated. Use use_memoise instead.
cache.path	Deprecated. Use cache_path instead.
do.parallel	Deprecated. Use do_parallel instead.

Value

A dataframe with requested datasets

Examples

```
## Not run:
my_id <- c('Selic Rate' = 11)
df <- gcbcb_get_series(my_id, cache_path = tempdir())

## End(Not run)
```

gcbcb_search_series	<i>Searches BCB-SGS series by text</i>
---------------------	--

Description

Searches the BCB-SGS catalog (sistema gerenciador de series temporais) for series whose name or metadata match a given text. The search is performed against the official BCB-SGS search service at <<https://www3.bcb.gov.br/sgspub/>>. The returned id column can be passed directly to [gcbcb_get_series](#).

Usage

```
gcbcb_search_series(
  search_text,
  max_results = 100,
  be_quiet = FALSE,
  use_memoise = TRUE,
  cache_path = gcbcb_get_default_cache_folder()
)
```

Arguments

search_text	A single string with the text to search for (e.g. 'selic').
max_results	Maximum number of series to return. Use Inf for no limit.
be_quiet	Logical. Should functions output messages to screen? - FALSE (default) or TRUE
use_memoise	Logical. Sets the use of caching system - TRUE (default) or FALSE
cache_path	Path to save cache files - 'gcbcb_cache' (default)

Value

A tibble with the columns `id`, `series_name`, `unit`, `frequency`, `first_date`, `last_date` and `source`. Returns zero rows when nothing matches.

Examples

```
## Not run:
df_search <- gcbcd_search_series('selic')
print(df_search)

# fetch the first matching series
my_id <- df_search$id[1]
names(my_id) <- df_search$series_name[1]
df_series <- gcbcd_get_series(my_id)

## End(Not run)
```

Index

gcbcd_get_series, [2](#), [3](#)
gcbcd_search_series, [3](#)