Package 'bacenR'

November 12, 2026

Title Access Data from Brazilian Central Bank: IFdata, Active Institutions, Balance Sheets and Normative Acts

Version 0.6.0

Description Provides functions to query, retrieve, and tidy economic and financial data from Brazilian Central Bank web services for use in R analyses and workflows. Active institutions information, bank branches, balance sheets, banking statistics by municipality (ESTBAN) and normative acts.

License GPL (>= 3)

URL https://github.com/rtheodoro/bacenR

BugReports https://github.com/rtheodoro/bacenR/issues

Depends R (>= 4.1.0)

Imports data.table, dplyr, fs, glue, httr, janitor, jsonlite, lifecycle, purrr, readr, readxl, rlang, stringr, tibble, tidyr, tools, utils, xml2

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

SystemRequirements qpdf

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ricardo Theodoro [aut, cre] (ORCID: <https://orcid.org/0000-0001-6741-5153>)

Maintainer Ricardo Theodoro <rtheodoro@usp.br>

Repository CRAN

Date/Publication 2026-11-12 14:50:14 UTC

Contents

get_balance_sheets 2
get_branches 4
get_estban 6
get_ifdata_registry 7
get_ifdata_reports 8
get_institutions 9
get_normative_data 11
get_normative_txt 12
tidy_balance_sheets 14
tidy_branches 16
tidy_estban 17
tidy_institutions 18

Index **20**

`get_balance_sheets` *Download balance sheets from* R*hrefhttps://www.bcb.gov.br/estabilidadefinanceira/balancetesbal*
Central Bank

Description

[Stable]

Downloads monthly "balancetes" (.CSV or .ZIP) released by the Brazilian Central Bank for one or more institution types and stores them in a local directory. ZIP files found in the destination directory are also extracted (errors during individual extractions are ignored).

Check the function `tidy_balance_sheets()` for tools to read and process

Usage

```
get_balance_sheets(
  institution,
  months = 12,
  first_year,
  final_year,
  out_dir,
  overwrite = FALSE
)
```

Arguments

`institution`	Character vector of institution types to download. Accepted values (case-insensitive): "BANCOS", "COOPERATIVAS", "CONSORCIOS" (Administradoras de consórcios), "CONGLOMERADOS" (Conglomerados financeiros), "SOCIEDADES", "BLOPRUDENCIAL" (Conglomerados Prudenciais), "COMBINADOS" (Combinados Cooperativos), "LIQUIDACAO" (Instituições em Regime Especial). The function maps these tokens to the type names used in the BCB URL.

months	Integer or integer vector specifying months to download. Values should be 1..12 (single values or vectors like c(6, 12)). When a month single-digit is provided it is zero-padded to two digits in filenames/URLs.
first_year	Integer, first year to download (inclusive). Defaults to 1993 (the oldest available balancete in the source).
final_year	Integer, last year to download (inclusive). Defaults to the previous calendar year.
out_dir	Character, directory where downloaded ZIP files (and extracted CSVs) will be written. The directory is created if it does not exist.
overwrite	Logical, whether to overwrite existing files in out_dir (default FALSE). If FALSE and a file already exists at the destination it is treated as a successful local result.

Details

For each combination of year, month and institution type the function constructs a URL of the form: `https://www.bcb.gov.br/content/estabilidadefinanceira/cosif/{tipo}/{YYYY}{MM}{institution}.CSV.ZIP` and attempts to download it with httr::GET. Successful and failed attempts are recorded in the returned tibble. After attempting all downloads the function tries to unzip any .zip files found in out_dir; unzip errors for individual files are caught and ignored so the process continues.

Invalid institution tokens cause an immediate error listing the invalid entries.

Value

A tibble with one row per attempted download and the following columns:

- years: Year attempted (integer).
- months: Month attempted (integer).
- institution: Institution token used (character).
- url: The URL attempted (character).
- dest: Destination file path for the downloaded ZIP (character).
- success: Logical indicating if the download (or existing-file fallback) was considered successful.
- http_status: HTTP status code returned by the request (integer NA if not available).
- error: Character string with an error message when success is FALSE, or a short note when a local file existed and was not overwritten.

References

Source Banco Central do Brasil (Bacen): https://www.bcb.gov.br/estabilidadefinanceira/balancetesbalancospatrimoniais Note: The values are not changed or processed in any way, they are downloaded as-is from the source.

Examples

```
# Download balance sheets of credit unions for December of 2023
get_balance_sheets(
  institution = "COOPERATIVAS",
  months = 12,
  first_year = 2023,
  final_year = 2023,
  out_dir = tempdir(),
  overwrite = FALSE
)

# Download balance sheets of banks and credit unions for June and December
# of 2022 and 2023
get_balance_sheets(
  institution = c("BANCOS", "COOPERATIVAS"),
  months = c(6, 12),
  first_year = 2022,
  final_year = 2023,
  out_dir = tempdir(),
  overwrite = FALSE
)
```

---

get_branches *Download the Lists of Branches and Service Posts from Bacen*

Description

[Experimental] Downloads the monthly lists of dependencies of institutions authorized by the Brazilian Central Bank (Bacen), from the Bacen website: branches (agências), service posts (postos de atendimento), electronic service posts (PAE) and branches of consortium administrators. Each file lists the dependencies in operation, with the full CNPJ (root, sequential number and check digits), address, municipality and opening date.

Usage

```
get_branches(
  type = "AGENCIAS",
  years,
  months,
  out_dir,
  cleanup_zip = TRUE,
  verbose = TRUE
)
```

Arguments

type	Character vector. Type(s) of dependency to download. Valid options are:

- "AGENCIAS": branches (since September 2007)
- "POSTOS": service posts (since September 2007)
- "PAE": electronic service posts (since July 2016)
- "FILIAISCONS": branches of consortium administrators (since July 2016)

Default is "AGENCIAS". Case-insensitive.

`years` Numeric vector. Year(s) to download, e.g. `c(2020:2023)`.

`months` Numeric vector. Month(s) to download, values between 1 and 12.

`out_dir` Character. Directory where the files are saved. Created if it does not exist.

`cleanup_zip` Logical. If TRUE, removes ZIP files after extraction. Default is TRUE.

`verbose` Logical. If TRUE, displays progress messages. Default is TRUE.

Details

Files are downloaded from `https://www.bcb.gov.br/fis/info/cad/<folder>/<YYYYMM><TYPE>.zip` and extracted to `out_dir/<YYYYMM>_<TYPE>/`. Use `tidy_branches()` to read them.

Notes from Bacen: from July 2016, the list of branches includes segments other than banks, and the list of service posts includes cooperative service posts (PAC), gold purchase posts (PCO) and temporary exchange posts (PCT) and no longer includes electronic posts (PAE), which have their own list.

Until June 2016, electronic posts are in the list of service posts (`tipo_dependencia`); a continuous series of PAE combines both lists. The December 2007 list of branches is named `200712AG.zip` and is handled automatically.

The branch totals match the branch counts reported in ESTBAN (`get_estban()`) for banks. Service posts and electronic posts have no CNPJ of their own; from 2020 they carry an installation id (`id_instalacao`).

Value

Invisibly, a tibble with one row per type and year-month requested (`type`, `year_month`, `url`) and the folder where the file was extracted (`path`, NA when the file is not available).

See Also

`tidy_branches()`, `get_estban()`

Examples

```
get_branches(type = "AGENCIAS", years = 2023, months = 12, out_dir = tempdir())
```

`get_estban` *Download ESTBAN (Banking Statistics by Municipality) Files from Bacen*

Description

[Experimental] Downloads the monthly ESTBAN files (Estatística Bancária Mensal por Município) from the Brazilian Central Bank (Bacen) website. Each file reports, for every municipality and institution, the number of branches (`agen_processadas`) and the balances of the COSIF accounts (`verbete_*`) of commercial banks, multiple banks with a commercial portfolio and Caixa Econômica Federal.

Usage

```
get_estban(years, months, out_dir, cleanup_zip = TRUE, verbose = TRUE)
```

Arguments

`years`	Numeric vector. Year(s) to download, e.g. `c(2020:2023)`.
`months`	Numeric vector. Month(s) to download, values between 1 and 12.
`out_dir`	Character. Directory where the files are saved. Created if it does not exist.
`cleanup_zip`	Logical. If TRUE, removes ZIP files after extraction. Default is TRUE.
`verbose`	Logical. If TRUE, displays progress messages. Default is TRUE.

Details

The file name changed over the years (`YYYYMM_ESTBAN.csv.zip`, `YYYYMM_ESTBAN.ZIP`, `YYYYMM_ESTBAN.csv`); each one is tried in turn, and error pages served by the website with status 200 are discarded. Files are saved as `YYYYMM_ESTBAN.csv` in `out_dir`. Use `tidy_estban()` to read them.

Value

Invisibly, a character vector with the paths of the extracted CSV files.

See Also

`tidy_estban()`

Examples

```
get_estban(years = 2023, months = 12, out_dir = tempdir())
```

get_ifdata_registry *Download institution registry data from* R*hrefhttps://olinda.bcb.gov.br/olinda/servico/IFDATA/versao/v1/aplicacaoIFdata of Brazilian Central Bank (Bacen) Cadastro*

Description

[Stable]

Download institution registry data from the Brazilian Central Bank (Bacen) IFdata API for specified years and months. The function handles multiple combinations of parameters and returns a consolidated data frame with the results.

Usage

```
get_ifdata_registry(year, month, verbose = TRUE)
```

Arguments

year	Numeric or vector. Year(s) to download (e.g., 2024 or c(2023, 2024))
month	Numeric or vector. Month(s) to download (1 to 12)
verbose	Logical. If TRUE, prints progress messages (default: TRUE)

Value

Data frame with institution registry data or NULL in case of error

Examples

```
# Single period
data <- get_ifdata_registry(
 year = 2024,
 month = 12
)
# Multiple years and months (all combinations), take some time to run
data <- get_ifdata_registry(
 year = c(2023, 2024),
 month = c(3, 6, 9, 12)
)
```

get_ifdata_reports *Download institution reports data from* R*hrefhttps://olinda.bcb.gov.br/olinda/servico/IFDATA/versao/v1/aplicacaoIFdata of Brazilian Central Bank (Bacen) Relatorios*

Description

[Stable]

Download IFdata reports from the Brazilian Central Bank (Bacen) API for specified years, months, institution types, and report types. The function handles multiple combinations of parameters and returns a consolidated data frame with the results.

Also, check the page to see the available reports for each institution type and date: IFdata.

Usage

```
get_ifdata_reports(year, month, type_institution, report, verbose = TRUE)
```

Arguments

`year`	Numeric or vector. year (ex: 2024 or c(2023, 2024))
`month`	Numeric or vector. month (ex: 1 to 12, or c(6, 12))
`type_institution`	Numeric. Type of Institution: • 1 = Conglomerados Prudenciais e Instituicoes Independentes • 2 = Conglomerados Financeiros e Instituicoes Independentes • 3 = Instituicoes Individuais • 4 = Instituicoes com Operacoes de Cambio
`report`	Numeric. Report type: • 1 = Resumo • 2 = Ativo • 3 = Passivo • 4 = Demonstracao de Resultado • 5 = Informacoes de Capital • 6 = Segmentacao • 7 = Carteira de Credito Ativa - Por indexador • 8 = Carteira de Credito ativa - por nivel de risco da operacao • 9 = Carteira de Credito ativa - por regiao geografica • 10 = Carteira de Credito ativa - quantidade de clientes e de operacoes • 11 = Carteira de Credito ativa Pessoa Fisica - modalidade e prazo de vencimento • 12 = Carteira de Credito ativa Pessoa Juridica - por atividade economica (CNAE)

- 13 = Carteira de Credito ativa Pessoa Juridica - modalidade e prazo de vencimento
- 14 = Carteira de Credito ativa Pessoa Juridica - por porte do tomador
- 15 = Movimentacao de Cambio no Trimestre
- 16 = Carteira de Credito ativa - por carteiras de instrumentos financeiros

`verbose` Logical. If TRUE, print progress messages (default: TRUE)

Value

Data frame with IFdata values or NULL in the case of errors.

Examples

```
# Unique institution type for a specific period
data <- get_ifdata_reports(
 year = 2024,
 month = 12,
 report = 1,
 type_institution = 2
)

# Multiple periods, take some time to run
data <- get_ifdata_reports(
  year = c(2023, 2024),
  month = c(6, 12),
  report = 1,
  type_institution = 2
)
```

`get_institutions` *Download and Process Brazilian Financial Institutions Data from Bacen*

Description

[Stable]

This function downloads financial institutions data from the Brazilian Central Bank (Bacen) website for specified institution types and date ranges. The data is downloaded as ZIP files, extracted, and can be optionally cleaned up.

Usage

```
get_institutions(
  institution,
  years,
  months,
  out_dir,
```

```
  cleanup_zip = TRUE,
  verbose = TRUE
)
```

Arguments

`institution`	Character vector. Type(s) of financial institutions to download. Valid options are: • "CONGLOMERADOS" • "BANCOS" • "COOPERATIVAS" • "ADMCONSORCIO" • "SOCIEDADES" Default is "COOPERATIVAS". Case-insensitive. Check the details on Bacen's website.
`years`	Numeric vector. Year(s) to download, e.g. `c(2020:2023)` or `c(2020, 2022)`.
`months`	Numeric vector. Month(s) to download, values between 1 and 12, e.g. `c(1:12)` or `c(6, 12)`.
`out_dir`	Character. Directory path where downloaded files will be saved. Default is "data". The directory will be created if it doesn't exist.
`cleanup_zip`	Logical. If TRUE, removes ZIP files after extraction. Default is TRUE.
`verbose`	Logical. If TRUE, displays progress messages and warnings. Default is TRUE.

Details

The function performs the following steps:

- Validates institution types against known valid options
- Generates every combination of `years` x `months` as year-month strings
- Downloads ZIP files for each institution and month from BCB website
- Extracts the downloaded ZIP files to the output directory
- Optionally removes ZIP files after extraction
- Displays progress information if verbose = TRUE

For `CONGLOMERADOS`, the file suffix on BCB's server changed from the plural "CONGLOMERADOS" to the singular "CONGLOMERADO" starting in months after 2022 (e.g. `.../202208CONGLOMERADOS.zip` vs. `.../202607CONGLOMERADO.zip`). This is handled internally and is transparent to the caller.

Institution type mappings:

- CONGLOMERADOS: Conglomerados
- BANCOS: Bancos comerciais, múltiplos e caixa
- COOPERATIVAS: Cooperativas de crédito
- ADMCONSORCIO: Consórcios administrativos
- SOCIEDADES: Bancos de investimentos, desenvolvimento e sociedades corretoras

Value

Invisible NULL. The function is called for its side effects (downloading and extracting files to the specified directory).

See Also

https://www.bcb.gov.br/estabilidadefinanceira/relacao_instituicoes_funcionamento

Examples

```
# Download cooperative credit unions data for 2023
get_institutions(
  institution = "COOPERATIVAS",
  years = 2023,
  months = 11:12,
  out_dir = tempdir()
)

# Download multiple institution types
get_institutions(
  institution = c("BANCOS", "COOPERATIVAS"),
  years = 2022,
  months = 1:12,
  out_dir = tempdir()
)

# Skip downloading, just use existing files
get_institutions(
  institution = "BANCOS",
  years = 2022,
  months = 1:12,
  out_dir = tempdir(),
  verbose = FALSE
)
```

---

get_normative_data *Download* R*hrefhttps://www.bcb.gov.br/estabilidadefinanceira/buscanormasnormative acts from the Brazilian Central Bank (Bacen) by terms and date range*

Description

[Stable]

Queries the Bacen normative search API, collects all results within the provided date range and returns a data.frame with the records. Data are downloaded from https://www.bcb.gov.br/estabilidadefinanceira/buscanormas.

Usage

```
get_normative_data(terms, ini_date, end_date)
```

Arguments

terms	Character vector. Search terms; they will be concatenated with " OR " and URL-encoded.
ini_date	Character. Start date in "YYYY-MM-DD" format.
end_date	Character. End date in "YYYY-MM-DD" format.

Details

1. Builds the query URL using the terms and the date range.
2. Retrieves the total number of results to determine the page size.
3. Iterates over result pages (incrementing by 500) until a page returns fewer than 13 rows (stop condition).
4. Performs simple cleaning on returned fields:
   - removes "string;#" prefixes,
   - removes HTML tags in summaries and subjects,
   - removes the decimal part of normative numbers.

Value

data.frame containing the records returned by the API and the post-processed columns. Returns an empty data.frame if the server is unavailable or no internet connection.

Examples

```
## Not run:
ini_date <- "2025-01-01"
end_date <- "2025-01-05"
terms <- c("Cooperativas", "Cooperativa")
normas <- get_normative_data(terms, ini_date, end_date)

## End(Not run)
```

---

get_normative_txt *Download* R*hrefhttps://www.bcb.gov.br/estabilidadefinanceira/buscanormasnormative texts from the Brazilian Central Bank (Bacen) from a reference data.frame*

Description

[Stable]

For each row of the input data.frame, constructs the appropriate URL for the Bacen normatives API, performs the request, extracts the returned content and applies basic cleaning to the text fields (Assunto and Texto). Returns a data.frame with all aggregated contents.

Usage

```
get_normative_txt(normative_data)
```

Arguments

normative_data	data.frame. A data.frame with at least the columns:

- TipodoNormativoOWSCHCS: type of the normative (e.g., "Comunicado", "Ato de Diretor", etc.)
- NumeroOWSNMBR: number/identifier of the normative to be queried.
- Typically the result of a prior query (e.g., via get_normative_data).

Details

The function performs the following steps:

1. Iterates over each row of normative_data (prints the iteration index for monitoring).
2. Encodes the normative type for URL usage (using URLencode(..., reserved = TRUE)).
3. Chooses the API route depending on the type:
   - for "Comunicado", "Ato de Diretor" and "Ato do Presidente" it uses the other-norms endpoint (exibeoutrasnormas);
   - for the rest, it uses the standard normative endpoint (exibenormativo).
4. Makes the HTTP request (via httr::GET) and converts the returned JSON into a data.frame.
5. Extracts and cleans textual fields:
   - converts HTML to plain text for the fields Assunto and Texto (using xml2::read_html + xml2::xml_text),
   - removes line breaks and extra spaces (stringr::str_replace_all, stringr::str_squish).
6. Aggregates each result into a final data.frame returned by the function.

Value

data.frame with the records returned by the API for each reference in normative_data, including the extracted and post-processed fields (e.g., Assunto, Texto).

Note

- The function assumes the BCB API returns a conteudo field with the expected fields.
- Request errors or changes in the API structure may cause failures; consider handling exceptions around the call if needed.
- Packages used internally: httr, jsonlite, xml2, stringr, dplyr, glue, purrr.

Examples

```
# First, download normative data
normative_data <- get_normative_data(
   "Cooperativa",
    "2023-08-01",
    "2023-08-10")
# Then, download the full texts for the retrieved normatives
normative_txt <- get_normative_txt(normative_data)
```

`tidy_balance_sheets` *Process and reshape balances CSV exports from the Brazilian Central Bank (Bacen) downloaded via* `get_balance_sheets()`.

Description

[Stable]

`tidy_balance_sheets` reads raw CSV export files produced by the BCB and downloaded via `get_balance_sheets()`, normalizes filenames, locates the header row, imports rows matching a specific document number, cleans column names, pivots account balances so that each account becomes a column and each row corresponds to an institution/date, and optionally writes per-institution-type CSVs.

Usage

```
tidy_balance_sheets(path_raw, out_dir, doc_filter = 4010, save = TRUE)
```

Arguments

`path_raw`	Character scalar. Directory containing the raw CSV files. Filenames are expected to start with a YYYYMM prefix (e.g. "202012COOPERATIVAS.CSV"). The function will look for files matching "\.CSV$" (case-insensitive).
`out_dir`	Character scalar. Directory where output CSVs will be saved when `save = TRUE`. If it does not exist it will be created.
`doc_filter`	Numeric or integer scalar. The numeric value used to filter rows in the imported tables by the column DOCUMENTO (e.g. 4010). Other options are: 4010, 4413, 4423, 4433, 4060, 4016 and 4066. Check the webpage for more details: BCB balance page
`save`	Logical scalar. If `TRUE` (default) the processed data frames are also written to disk as CSV files (one per institution type). If `FALSE` the function only returns the processed data.frames.

Details

Behavior and implementation notes:

- Filenames are first normalized by removing all underscores. If a target filename after normalization already exists, the function aborts to avoid overwriting files.
- The function extracts year/month and an institution type token from the file basename by assuming the first six characters are YYYYMM and the remaining characters (without the ".CSV") represent the type.
- For each file the function reads the file as text and attempts to locate the header row by scanning lines until it finds a row that contains (case-insensitive) tokens matching:
  - "DOCUMENTO" (or "DOCUMENTOS"),
  - "CNPJ",
  - and "DATA BASE" (variants matched by regex such as "DATA_BASE" or "#DATA_BASE"). The delimiter for that header line is heuristically detected among semicolon (";"), comma (","), or whitespace. For whitespace-delimited data the function uses read.table's default behavior.
- After locating the header, the table is read with `utils::read.table` using `header = TRUE`, the detected separator, latin1 encoding, and `check.names = FALSE`. Rows where the DOCUMENTO column equals `doc_filter` are kept.
- Column names are cleaned via `janitor::clean_names()` before the wide pivot. The function expects at least the following cleaned column names to be present: `conta`, `saldo`, and `data`. The pivot uses the id columns (`data`, `documento`, `cnpj`, `nome_instituicao`), `conta` as names and `saldo` as values.
- If `save = TRUE` each resulting data.frame is written to CSV using `readr::write_csv()` to the path returned by `output_filename(type_inst)` (wrapped with `out_dir`).
- The function will stop with informative errors in several situations: missing CSVs in `path_raw`, failure to find a header line with the required tokens, missing required columns after import/cleaning, or failure to rename files when removing underscores.

Value

A named list of data.frames (one element per unique institution type detected in the filenames). Each data.frame has one row per unique combination of (data, documento, cnpj, nome_instituicao) and one column per account (the "conta" values) with balances populated from the "saldo" column.

Note

The function prints a message reminding that the numeric values in the balance sheets are not adjusted for inflation. See: https://www.bcb.gov.br/estabilidadefinanceira/balancetesbalancospatrimoniais

Examples

```
# First, download balance sheets
dir <- file.path(tempdir(), "balance_sheets")
get_balance_sheets(
  institution = "BANCOS",
  months = 12,
```

```
  first_year = 2023,
  final_year = 2023,
  out_dir = dir,
  overwrite = FALSE
)

# Now, tidy the files
# Do not write outputs
tidy_balance_sheets(
  path_raw = dir,
  out_dir = dir,
  doc_filter = 4010,
  save = FALSE
)

# Write outputs
tidy_balance_sheets(
  path_raw = dir,
  out_dir = file.path(tempdir(), "balance_sheets_out"),
  doc_filter = 4016,
  save = TRUE
)
```

`tidy_branches` *Read and Tidy the Lists of Branches and Service Posts*

Description

[Experimental] Reads the files downloaded with `get_branches()` and combines them into a single table, one row per type, year-month and dependency.

Usage

```
tidy_branches(path_dir, type = NULL, out_dir = NULL, verbose = TRUE)
```

Arguments

`path_dir`	Character. Directory with the folders created by `get_branches()` (`<YYYYMM>_<TYPE>/`).
`type`	Character vector or NULL. Type(s) to read (see `get_branches()`); NULL reads all. Default is NULL.
`out_dir`	Character or NULL. If given, the table is also written to `out_dir/<TYPE>.csv`, one file per type. Default is NULL.
`verbose`	Logical. If TRUE, displays progress messages. Default is TRUE.

Details

Every spreadsheet and every sheet in each folder is read, and the header row is the first one containing `CNPJ`. In the lists of service posts and electronic posts, the `CNPJ` column holds only the root of the institution, the post is identified by `id_instalacao`, and `tipo_dependencia` gives the kind of post (PAB, PAA, PAC, ...); these lists have no opening date. Dependencies abroad (branches without the CNPJ sequential number, posts without a state) are dropped with a message. Rows repeated within a type and year-month (same CNPJ or `id_instalacao`) are kept once.

Value

A tibble with `type`, `year_month`, `cnpj` (14 digits; NA for service posts and electronic posts, which have no CNPJ of their own and are identified by `id_instalacao`), `cnpj_basico`, `cnpj_ordem`, `cnpj_dv` and the remaining columns of the files in snake case. Columns whose names changed over the years are harmonized: `nome_instituicao`, `segmento`, `compe_banco`, `compe_agencia`, `nome_agencia`, `endereco`, `numero`, `complemento`, `bairro`, `cep`, `municipio`, `uf`, `codmun_ibge`, `data_inicio` (Date) and `id_instalacao`.

See Also

`get_branches()`

Examples

```
dir <- file.path(tempdir(), "branches")
get_branches(type = c("AGENCIAS", "POSTOS"), years = 2023, months = 12, out_dir = dir)
branches <- tidy_branches(dir)
```

---

tidy_estban *Read and Tidy ESTBAN Files*

Description

[Experimental] Reads the ESTBAN files downloaded with `get_estban()` and combines them into a single table, one row per year-month, municipality and institution.

Usage

```
tidy_estban(path_dir, out_dir = NULL, verbose = TRUE)
```

Arguments

`path_dir`	Character. Directory with the `YYYYMM_ESTBAN.csv` files (searched recursively).
`out_dir`	Character or NULL. If given, the combined table is also written to `out_dir/ESTBAN.csv`. Default is NULL.
`verbose`	Logical. If TRUE, displays progress messages. Default is TRUE.

Details

The header line starts with `#DATA_BASE` and is located automatically; values use a comma as decimal mark and the files are Latin-1 encoded. The column with the IBGE municipality code (`codmun_ibge`) exists only in the more recent files; older files identify the municipality by name and state.

Value

A tibble with the columns of the files in snake case: `data_base`, `uf`, `codmun_ibge` (when available in the file), `municipio`, `cnpj`, `nome_instituicao`, `agencias` (branches processed), `agen_esperadas` and the `verbete_*` balances as numeric, plus `year_month` (`YYYYMM`).

See Also

`get_estban()`

Examples

```
dir <- file.path(tempdir(), "estban")
get_estban(years = 2023, months = 12, out_dir = dir)
estban <- tidy_estban(dir)
```

---

tidy_institutions *Process and unify institution CSV exports from the Brazilian Central Bank (Bacen) downloaded via* `get_institutions()`.

Description

[Stable]

This function reads and processes financial institutions data from the Brazilian Central Bank (Bacen) files that were previously downloaded using the `get_institutions()` function. It takes a directory path containing the institution files, processes them, and optionally saves the output to a specified directory.

Usage

```
tidy_institutions(path_dir, out_dir, verbose = TRUE)
```

Arguments

`path_dir`	Character string specifying the path to the directory containing institution files to be processed.
`out_dir`	Character string specifying the path to the directory where processed output files should be saved.
`verbose`	Logical value indicating whether to print progress messages during processing. Default is typically FALSE.

Details

Bacen changes column headers over the years. After `janitor::clean_names()`, columns are renamed to the most recent header so each institution type has a single column per field:

- All types: `nome`, `nomeinstituicao` -> `nome_instituicao`; `fone` -> `telefone`; `cidade` -> `municipio`; `segmentos` -> `segmento`.
- E-mail and website columns (`email`, `e_mail`, `endereco_eletronico`, `sitio_na_internet`, `end_pagina_internet`) are assigned to `e_mail` or `sitio_na_internet` by their content (share of values containing "@"), because Bacen's headers for them are inconsistent and sometimes swapped.
- COOPERATIVAS: `classe_cooper` -> `classe`.
- CONGLOMERADOS: `nome_conglomerado` -> `nome_do_conglomerado`; `classe` -> `classe_do_conglomerado`; `nome_instituicao_participante`, `nome_participante` -> `nome_do_participante`; `dt_inicio` -> `data_inicio` (converted to `Date`); `estado` -> `uf`.

Columns that are not equivalent across years (e.g. COOPERATIVAS `tipo_cooperativa` until 2015 and `criterio_de_associacao` from 2016) are kept separate.

Value

A list of data.frames, where each element corresponds to a processed institution file. The names of the list elements typically correspond to institution identifiers or file names.

Examples

```
# First, download institution data
# Download cooperative credit unions data for 2023
get_institutions(
  institution = "COOPERATIVAS",
  years = 2023,
  months = 11:12,
  out_dir = tempdir()
)
# Process institution files from a directory
institutions <- tidy_institutions(
  path_dir = tempdir(),
  out_dir = tempdir(),
  verbose = TRUE
)
```

Index

get_balance_sheets, 2
get_branches, 4
get_branches(), *16, 17*
get_estban, 6
get_estban(), *5*, *17, 18*
get_ifdata_registry, 7
get_ifdata_reports, 8
get_institutions, 9
get_normative_data, 11
get_normative_txt, 12

tidy_balance_sheets, 14
tidy_branches, 16
tidy_branches(), *5*
tidy_estban, 17
tidy_estban(), *6*
tidy_institutions, 18