Package 'pixr'

May 9, 2026

Title Access Brazilian Central Bank 'PIX' Open Data 'API'

Version 0.1.0

Description Provides a 'tidyverse'-style interface to the Brazilian Central Bank (<https://www.bcb.gov.br>) 'PIX' Open Data 'API' <https://olinda.bcb.gov.br/olinda/servico/Pix_DadosAbertos/versao/v1/aplicacao#!/recursos>. Retrieve statistics on 'PIX' keys, transactions by municipality, and monthly transaction summaries. All functions return 'tibbles' and support 'OData' query parameters for filtering, selecting, and ordering data.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports httr2 (>= 1.0.0), cli (>= 3.6.0), tibble (>= 3.2.0), dplyr (>= 1.1.0), rlang (>= 1.1.0), purrr (>= 1.0.0)

Suggests testthat (>= 3.0.0), withr, knitr, rmarkdown, ggplot2, scales, forcats, tidyr

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/StrategicProjects/pixr

BugReports https://github.com/StrategicProjects/pixr/issues

NeedsCompilation no

Author Andre Leite [aut, cre], Marcos Wasilew [aut], Hugo Vasconcelos [aut], Diogo Bezerra [aut]

Maintainer Andre Leite <leite@castlab.org>

Repository CRAN

Date/Publication 2026-01-31 19:00:08 UTC

Contents

`format_brl` *Format Currency Value*

Description

Formats a numeric value as Brazilian Real (BRL) currency.

Usage

```
format_brl(x, prefix = TRUE, decimal_mark = ",", big_mark = ".")
```

Arguments

`x`	Numeric vector.
`prefix`	Logical; if TRUE (default), includes "R$" prefix.
`decimal_mark`	Character to use as decimal separator.
`big_mark`	Character to use as thousands separator.

Value

A character vector with formatted currency values.

Examples

```
format_brl(1234567.89)
format_brl(c(1000, 2000, 3000))
```

`get_pix_fraud_stats` *Get PIX Fraud Statistics (MED)*

Description

Retrieves fraud statistics for PIX transactions reported through the Special Return Mechanism (MED - Mecanismo Especial de Devolução).

Usage

```
get_pix_fraud_stats(
  database,
  filter = NULL,
  columns = NULL,
  top = NULL,
  skip = NULL,
  orderby = NULL,
  verbose = TRUE
)
```

Arguments

`database`	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
`filter`	OData filter expression as a character string.
`columns`	Character vector of columns to return. If NULL, returns all columns.
`top`	Integer; maximum number of records to return.
`skip`	Integer; number of records to skip (for pagination).
`orderby`	Character string specifying the column to sort by. Use `"Column"` for ascending or `"Column desc"` for descending order.
`verbose`	Logical; if TRUE (default), prints progress messages.

Details

The MED (Mecanismo Especial de Devolução) is a mechanism created by the Brazilian Central Bank to facilitate the return of funds in cases of fraud or operational errors in PIX transactions.

This endpoint provides statistics on:

- Number of fraud reports (notificações de infração)
- Number of return requests (pedidos de devolução)
- Values involved in fraud cases

Value

A tibble::tibble with PIX fraud statistics.

Examples

```
## Not run: # It usually takes much longer than 5 seconds.
# Get fraud statistics for September 2025
fraud <- get_pix_fraud_stats(database = "202509")

# Get top 100 records
fraud <- get_pix_fraud_stats(database = "202509", top = 100)

## End(Not run)
```

`get_pix_fraud_stats_multi` *Get PIX Fraud Statistics for Multiple Months*

Description

Retrieves fraud statistics for multiple months and combines them into a single tibble.

Usage

```
get_pix_fraud_stats_multi(databases, ...)
```

Arguments

`databases`	Character vector of year-months in "YYYYMM" format.
`...`	Additional arguments passed to `get_pix_fraud_stats()`.

Value

A tibble::tibble with combined fraud statistics.

Examples

```
# It usually takes much longer than 5 seconds.
# Get fraud data for Q3 2025
q3_fraud <- get_pix_fraud_stats_multi(
  databases = c("202507", "202508", "202509")
)
```

get_pix_keys *Get PIX Keys Stock by Participant*

Description

Retrieves the stock of PIX keys registered in the Directory of Transactional Account Identifiers (DICT) at the end of each month, broken down by PIX participant and key type.

Usage

```
get_pix_keys(
  date,
  filter = NULL,
  columns = NULL,
  top = NULL,
  skip = NULL,
  orderby = NULL,
  verbose = TRUE
)
```

Arguments

date	Character string in "YYYY-MM-DD" format specifying the reference date. This parameter is **required**. The API returns data for the last day of the specified month.
filter	OData filter expression as a character string. Examples: • "NaturezaUsuario eq 'PF'" - Filter by user type (PF or PJ) • "TipoChave eq 'CPF'" - Filter by key type • "Nome eq 'BANCO DO BRASIL S.A.'" - Filter by institution name
columns	Character vector of columns to return. If NULL, returns all columns. See "Available Columns" section.
top	Integer; maximum number of records to return.
skip	Integer; number of records to skip (for pagination).
orderby	Character string specifying the column to sort by. Use "Column" for ascending or "Column desc" for descending order.
verbose	Logical; if TRUE (default), prints progress messages.

Details

The BCB PIX API requires a date parameter specifying which date's data to retrieve. The data shows the number of PIX keys registered by each financial institution (participant).

Note: The API returns data for the last day of the month containing the specified date. For example, date = "2025-12-01" returns data for 2025-12-31.

Value

A tibble::tibble with PIX keys data.

Available Columns

Data Reference date (last day of month, YYYY-MM-DD format)

ISPB 8-digit code identifying the financial institution

Nome Name of the PIX participant (financial institution)

NaturezaUsuario User type: PF (Individual) or PJ (Legal Entity)

TipoChave Key type: CPF, CNPJ, Celular, e-mail, or Aleatória

qtdChaves Number of registered keys

Examples

```
# It usually takes much longer than 5 seconds.
# Get all PIX keys data for December 2025
keys <- get_pix_keys(date = "2025-12-01")

# Filter by key type and order by quantity
cpf_keys <- get_pix_keys(
  date = "2025-12-01",
  filter = "TipoChave eq 'CPF'",
  orderby = "qtdChaves desc",
  top = 100
)

# Filter by institution
bb_keys <- get_pix_keys(
  date = "2025-12-01",
  filter = "Nome eq 'BANCO DO BRASIL S.A.'"
)
```

---

get_pix_keys_by_type *Get PIX Keys by Key Type*

Description

Retrieves PIX keys data and returns a summary by key type.

Usage

```
get_pix_keys_by_type(date, verbose = TRUE)
```

Arguments

date	Character string in "YYYY-MM-DD" format specifying the reference date. This parameter is **required**. The API returns data for the last day of the specified month.
verbose	Logical; if TRUE (default), prints progress messages.

Value

A tibble::tibble with summary data by key type.

Examples

```
# It usually takes much longer than 5 seconds.
# Get summary by key type
get_pix_keys_by_type(date = "2025-12-01")
```

---

get_pix_keys_summary *Get PIX Keys Summary by Institution*

Description

Retrieves PIX keys data and returns a summary showing total keys by institution, sorted by total keys.

Usage

```
get_pix_keys_summary(date, n_top = 20, verbose = TRUE)
```

Arguments

date	Character string in "YYYY-MM-DD" format specifying the reference date. This parameter is **required**. The API returns data for the last day of the specified month.
n_top	Integer; number of top institutions to return. Default is 20.
verbose	Logical; if TRUE (default), prints progress messages.

Value

A tibble::tibble with summary data by institution.

Examples

```
# It usually takes much longer than 5 seconds.
# Get top 20 institutions by total keys
get_pix_keys_summary(date = "2025-12-01")

# Get top 10 institutions
get_pix_keys_summary(date = "2025-12-01", n_top = 10)
```

`get_pix_summary` *Get PIX Transaction Summary*

Description

Retrieves transaction statistics and aggregates them by specified grouping variables. This is a convenience function that fetches data and performs common aggregations.

Usage

```
get_pix_summary(database, group_by = "NATUREZA", verbose = TRUE)
```

Arguments

`database`	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
`group_by`	Character vector of columns to group by. Common choices: "NATUREZA", "PAG_REGIAO", "REC_REGIAO", "FORMAINICIACAO".
`verbose`	Logical; if TRUE (default), prints progress messages.

Value

A tibble::tibble with aggregated transaction statistics.

Examples

```
# It usually takes much longer than 5 seconds.
# Summary by transaction nature
get_pix_summary(database = "202509", group_by = "NATUREZA")

# Summary by payer region
get_pix_summary(database = "202509", group_by = "PAG_REGIAO")

# Summary by nature and initiation method
get_pix_summary(database = "202509", group_by = c("NATUREZA", "FORMAINICIACAO"))
```

`get_pix_transactions_by_municipality` *Get PIX Transactions by Municipality*

Description

Retrieves PIX transaction statistics aggregated by municipality, showing the number and value of transactions from the perspective of both payers and receivers.

Usage

```
get_pix_transactions_by_municipality(
  database,
  filter = NULL,
  columns = NULL,
  top = NULL,
  skip = NULL,
  orderby = NULL,
  verbose = TRUE
)
```

Arguments

`database`	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
`filter`	OData filter expression as a character string. Examples: • `"Estado eq 'SÃO PAULO'"` - Filter by state name • `"Sigla_Regiao eq 'SE'"` - Filter by region • `"Municipio eq 'RECIFE'"` - Filter by municipality name
`columns`	Character vector of columns to return. If NULL, returns all columns. See "Available Columns" section.
`top`	Integer; maximum number of records to return.
`skip`	Integer; number of records to skip (for pagination).
`orderby`	Character string specifying the column to sort by. Use `"Column"` for ascending or `"Column desc"` for descending order.
`verbose`	Logical; if TRUE (default), prints progress messages.

Details

The BCB PIX API requires a `database` parameter (YYYYMM format) specifying which month's data to retrieve. The data is broken down by municipality, person type (PF/PJ), and transaction direction (payer/receiver).

Value

A tibble::tibble with PIX transaction data by municipality.

Available Columns

AnoMes Reference year-month in YYYYMM format

Municipio_Ibge IBGE municipality code

Municipio Municipality name

Estado_Ibge IBGE state code

Estado State name

Sigla_Regiao Region abbreviation (NE, SE, S, CO, N)

Regiao Region name (NORDESTE, SUDESTE, SUL, CENTRO-OESTE, NORTE)

VL_PagadorPF Total value paid by individuals (BRL)

QT_PagadorPF Number of transactions where individuals were payers

VL_PagadorPJ Total value paid by legal entities (BRL)

QT_PagadorPJ Number of transactions where legal entities were payers

VL_RecebedorPF Total value received by individuals (BRL)

QT_RecebedorPF Number of transactions where individuals were receivers

VL_RecebedorPJ Total value received by legal entities (BRL)

QT_RecebedorPJ Number of transactions where legal entities were receivers

QT_PES_PagadorPF Number of distinct individual payers

QT_PES_PagadorPJ Number of distinct legal entity payers

QT_PES_RecebedorPF Number of distinct individual receivers

QT_PES_RecebedorPJ Number of distinct legal entity receivers

Examples

```
# It usually takes much longer than 5 seconds.
# Get municipality transaction data for December 2025
muni <- get_pix_transactions_by_municipality(database = "202512")

# Filter by state
maranhao <- get_pix_transactions_by_municipality(
  database = "202512",
  filter = "Estado eq 'MARANHÃO'",
  orderby = "Municipio desc",
  top = 10
)

# Filter by region
nordeste <- get_pix_transactions_by_municipality(
  database = "202512",
  filter = "Sigla_Regiao eq 'NE'"
)

# Order by value
top_value <- get_pix_transactions_by_municipality(
  database = "202512",
```

```
  orderby = "VL_PagadorPF desc",
  top = 100
)
```

`get_pix_transactions_by_region` *Get PIX Transactions by Region*

Description

A convenience wrapper that aggregates municipality data at the region level.

Usage

```
get_pix_transactions_by_region(database, verbose = TRUE)
```

Arguments

`database`	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
`verbose`	Logical; if TRUE (default), prints progress messages.

Value

A tibble::tibble with PIX transaction data aggregated by region.

Examples

```
# It usually takes much longer than 5 seconds.
# Get region-level aggregates
regions <- get_pix_transactions_by_region(database = "202512")
```

`get_pix_transactions_by_state` *Get PIX Transactions by State*

Description

A convenience wrapper around `get_pix_transactions_by_municipality()` that aggregates data at the state level.

Usage

```
get_pix_transactions_by_state(database, verbose = TRUE)
```

Arguments

database	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
verbose	Logical; if TRUE (default), prints progress messages.

Value

A tibble::tibble with PIX transaction data aggregated by state.

Examples

```
# It usually takes much longer than 5 seconds.
# Get state-level aggregates for December 2025
states <- get_pix_transactions_by_state(database = "202512")
```

---

get_pix_transaction_stats

Get PIX Transaction Statistics

Description

Retrieves detailed statistics on PIX transactions settled through the Instant Payment System (SPI), with breakdowns by payer/receiver type, region, age group, initiation method, and transaction nature.

Usage

```
get_pix_transaction_stats(
  database,
  filter = NULL,
  columns = NULL,
  top = NULL,
  skip = NULL,
  orderby = NULL,
  verbose = TRUE
)
```

Arguments

database	Character string in "YYYYMM" format specifying which month's data to retrieve. This parameter is **required**.
filter	OData filter expression as a character string. Examples: • `"NATUREZA eq 'P2P'"` - Filter by transaction nature • `"PAG_REGIAO eq 'SUDESTE'"` - Filter by payer region • `"FORMAINICIACAO eq 'DICT'"` - Filter by initiation method

`columns`	Character vector of columns to return. If NULL, returns all columns. See "Available Columns" section.
`top`	Integer; maximum number of records to return.
`skip`	Integer; number of records to skip (for pagination).
`orderby`	Character string specifying the column to sort by. Use `"Column"` for ascending or `"Column desc"` for descending order.
`verbose`	Logical; if TRUE (default), prints progress messages.

Details

The BCB PIX API requires a `database` parameter specifying which month's data to retrieve. The data provides granular breakdowns of PIX transactions.

Transaction Nature Codes:

- **P2P**: Person to Person
- **P2B**: Person to Business
- **B2P**: Business to Person
- **B2B**: Business to Business
- **P2G**: Person to Government
- **G2P**: Government to Person

Initiation Methods:

- **DICT**: PIX Key lookup
- **QRDN**: Dynamic QR Code
- **QRES**: Static QR Code
- **MANU**: Manual entry (bank details)
- **INIC**: Payment Initiator

Value

A tibble::tibble with PIX transaction statistics.

Available Columns

AnoMes Reference year-month as integer (YYYYMM format)

PAG_PFPJ Payer type: PF (Individual) or PJ (Legal Entity)

REC_PFPJ Receiver type: PF (Individual) or PJ (Legal Entity)

PAG_REGIAO Payer region (NORTE, NORDESTE, SUDESTE, SUL, CENTRO-OESTE)

REC_REGIAO Receiver region

PAG_IDADE Payer age group

REC_IDADE Receiver age group

FORMAINICIACAO Initiation method (DICT, QRDN, QRES, MANU, INIC)

NATUREZA Transaction nature (P2P, P2B, B2P, B2B, P2G, G2P)

FINALIDADE Transaction purpose (Pix, Pix Saque, Pix Troco, etc.)

VALOR Total transaction value (in BRL)

QUANTIDADE Number of transactions

Examples

```
# It usually takes much longer than 5 seconds.
# Get transaction statistics for September 2025
stats <- get_pix_transaction_stats(database = "202509")

# Filter by transaction nature
p2p <- get_pix_transaction_stats(
  database = "202509",
  filter = "NATUREZA eq 'P2P'"
)

# Filter by region and order by value
sudeste <- get_pix_transaction_stats(
  database = "202509",
  filter = "PAG_REGIAO eq 'SUDESTE'",
  orderby = "VALOR desc",
  top = 100
)

# Multiple filters (use 'and')
filtered <- get_pix_transaction_stats(
  database = "202509",
  filter = "NATUREZA eq 'P2P' and PAG_REGIAO eq 'NORDESTE'"
)
```

---

get_pix_transaction_stats_multi

Get PIX Transaction Statistics for Multiple Months

Description

Retrieves transaction statistics for multiple months and combines them into a single tibble.

Usage

```
get_pix_transaction_stats_multi(databases, ...)
```

Arguments

databases	Character vector of year-months in "YYYYMM" format.
...	Additional arguments passed to `get_pix_transaction_stats()`.

Value

A tibble::tibble with combined transaction statistics.

Examples

```
# It usually takes much longer than 5 seconds.
# Get data for Q3 2025
q3_data <- get_pix_transaction_stats_multi(
  databases = c("202507", "202508", "202509")
)
```

`pix_columns` *Get Column Information for a PIX Endpoint*

Description

Returns detailed information about the columns available for a specific endpoint.

Usage

```
pix_columns(endpoint = c("keys", "municipality", "stats", "fraud"))
```

Arguments

`endpoint`	Character string specifying the endpoint. One of: "keys", "municipality", "stats", or "fraud".

Value

A tibble::tibble with column names, types, and descriptions.

Examples

```
pix_columns("keys")
pix_columns("municipality")
pix_columns("stats")
pix_columns("fraud")
```

`pix_endpoints` *Get Available PIX API Endpoints*

Description

Returns information about all available endpoints in the BCB PIX Open Data API.

Usage

```
pix_endpoints()
```

Value

A tibble::tibble with endpoint names, descriptions, parameters, and associated functions.

Examples

```
pix_endpoints()
```

`pix_ping` *Check API Connection*

Description

Tests the connection to all BCB PIX Open Data API endpoints. Each endpoint is tested with a single record request (top=1).

Usage

```
pix_ping()
```

Value

A tibble (invisibly) with columns:

endpoint Name of the endpoint tested

status Result: "OK" or error message

time_seconds Time taken for the request in seconds

Examples

```
## Not run: # It usually takes much longer than 5 seconds.
# Test all endpoints
pix_ping()

# Capture results
results <- pix_ping()
print(results)

## End(Not run)
```

pix_query *Get Raw Data from PIX API*

Description

Low-level function to fetch data from any PIX API endpoint with custom parameters.

Usage

```
pix_query(
  endpoint,
  params = NULL,
  filter = NULL,
  select = NULL,
  orderby = NULL,
  top = NULL,
  skip = NULL,
  format = "json",
  verbose = TRUE
)
```

Arguments

endpoint	Character string specifying the endpoint name.
params	Named list of endpoint parameters. Each endpoint requires different parameters: • ChavesPix: `list(Data = "YYYY-MM-DD")` • TransacoesPixPorMunicipio: `list(DataBase = "YYYYMM")` • EstatisticasTransacoesPix: `list(Database = "YYYYMM")` • EstatisticasFraudesPix: `list(Database = "YYYYMM")`
filter	OData filter expression as a character string.
select	Character vector of columns to select.
orderby	OData orderby expression as a character string.
top	Integer; maximum number of records to return.
skip	Integer; number of records to skip.
format	Response format: "json" (default), "xml", "csv", or "html".
verbose	Logical; if TRUE, prints progress messages.

Value

A tibble::tibble with the raw API response data.

Examples

```
## Not run: # It usually takes much longer than 5 seconds.
# Custom query for keys
pix_query(
  endpoint = "ChavesPix",
  params = list(Data = "2025-12-01"),
  top = 10
)

# Custom query for transaction stats
pix_query(
  endpoint = "EstatisticasTransacoesPix",
  params = list(Database = "202509"),
  top = 10
)

## End(Not run)
```

`pix_timeout` *Get or Set API Request Timeout*

Description

Get or set the timeout for API requests. The default timeout is 120 seconds. The BCB API can be slow for large queries, so a generous timeout is recommended.

Usage

```
pix_timeout(seconds = NULL)
```

Arguments

`seconds`	Integer; timeout in seconds. If NULL, returns the current timeout.

Value

- `pix_timeout()`: Returns the current timeout in seconds (invisibly when setting).
- When called with `seconds`, sets the timeout and returns the new value invisibly.

Examples

```
# Get current timeout
pix_timeout()

# Set timeout to 180 seconds (3 minutes)
pix_timeout(180)

# Set timeout to 60 seconds
pix_timeout(60)
```

```
# Reset to default (120 seconds)
pix_timeout(120)
```

`pix_url` *Build PIX API URL (for debugging)*

Description

Builds and returns the URL that would be called by a PIX API function. Useful for debugging and testing.

Usage

```
pix_url(
  endpoint,
  params = NULL,
  filter = NULL,
  select = NULL,
  orderby = NULL,
  top = NULL,
  skip = NULL,
  format = "json"
)
```

Arguments

`endpoint`	Character string specifying the endpoint name.
`params`	Named list of endpoint parameters. Each endpoint requires different parameters: • ChavesPix: `list(Data = "YYYY-MM-DD")` • TransacoesPixPorMunicipio: `list(DataBase = "YYYYMM")` • EstatisticasTransacoesPix: `list(Database = "YYYYMM")` • EstatisticasFraudesPix: `list(Database = "YYYYMM")`
`filter`	OData filter expression as a character string.
`select`	Character vector of columns to select.
`orderby`	OData orderby expression (e.g., "Column asc" or "Column desc").
`top`	Integer; maximum number of records to return.
`skip`	Integer; number of records to skip.
`format`	Response format: "json" (default), "xml", "csv", or "html".

Value

Character string with the full API URL.

Examples

```
# See what URL would be called for each endpoint
pix_url("ChavesPix", params = list(Data = "2025-12-01"), top = 10)
pix_url("EstatisticasTransacoesPix", params = list(Database = "202509"), top = 10)
pix_url("TransacoesPixPorMunicipio", params = list(DataBase = "202512"), top = 10)
```

`year_month_to_date` *Convert Year-Month String to Date*

Description

Converts a year-month string in "YYYYMM" format to a Date object (first day of the month).

Usage

```
year_month_to_date(year_month)
```

Arguments

`year_month`	Character vector of year-month strings in "YYYYMM" format.

Value

A Date vector.

Examples

```
year_month_to_date("202312")
year_month_to_date(c("202301", "202302", "202303"))
```

Index