

# Package: tesouror (via r-universe)

June 6, 2026

**Title** Access Brazilian National Treasury Open Data APIs

**Version** 0.2.2

**Description** Provides a unified interface to access open data from the Brazilian National Treasury ('Tesouro Nacional') and related government APIs. Covers six data sources: 'SICONFI' <<https://apidatalake.tesouro.gov.br/docs/siconfi/>> for fiscal reports ('RREO', 'RGF', 'DCA', 'MSC') and entity information; 'CUSTOS' <<https://apidatalake.tesouro.gov.br/docs/custos/>> for federal government cost data; 'SADIPEM' <<https://apidatalake.tesouro.gov.br/docs/sadipem/>> for public debt and credit operations; 'Transferencias Constitucionais' <[https://apiapex.tesouro.gov.br/aria/v1/transferencias\\_constitucionais/docs](https://apiapex.tesouro.gov.br/aria/v1/transferencias_constitucionais/docs)> for constitutional transfers to states and municipalities; 'SIORG' <<https://estruturaorganizational.dados.gov.br>> for federal organizational structure; and 'SIOPE' ('FNDE'/MEC) for education spending data. Features automatic pagination, in-memory caching, retry logic, and tidy output.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.1.0)

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**Imports** cli (>= 3.6.0), dplyr (>= 1.1.0), httr2 (>= 1.0.0), janitor (>= 2.2.0), stringi (>= 1.7.0), stringr (>= 1.5.0), tibble (>= 3.2.0)

**Suggests** httpptest2 (>= 1.0.0), knitr, rmarkdown, testthat (>= 3.0.0), withr

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://github.com/StrategicProjects/tesouror>,  
<https://strategicprojects.github.io/tesouror/>

**BugReports** <https://github.com/StrategicProjects/tesouror/issues>

**Config/pak/sysreqs** libicu-dev libssl-dev

**Repository** <https://strategicprojects.r-universe.dev>

**Date/Publication** 2026-05-04 13:33:31 UTC

**RemoteUrl** <https://github.com/strategicprojects/tesouror>

**RemoteRef** HEAD

**RemoteSha** cc30df36e7c5dc6b619b3de7377d046dee5c0a53

## Contents

get_anexos . . . . .	3
get_custos_demais . . . . .	4
get_custos_depreciacao . . . . .	6
get_custos_pensionistas . . . . .	8
get_custos_pessoal_ativo . . . . .	10
get_custos_pessoal_inativo . . . . .	12
get_custos_transferencias . . . . .	14
get_dca . . . . .	16
get_dca_for_state . . . . .	17
get_entes . . . . .	19
get_extrato . . . . .	20
get_msc_controle . . . . .	21
get_msc_orcamentaria . . . . .	23
get_msc_patrimonial . . . . .	25
get_opc_cronograma_liberacoes . . . . .	26
get_opc_cronograma_pagamentos . . . . .	28
get_opc_taxa_cambio . . . . .	29
get_pvl . . . . .	30
get_pvl_tramitacao . . . . .	32
get_res_cdp . . . . .	33
get_res_cronograma_pagamentos . . . . .	34
get_rgf . . . . .	35
get_rgf_for_state . . . . .	37
get_rreo . . . . .	39
get_rreo_for_state . . . . .	41
get_siope_dados_gerais . . . . .	43
get_siope_despesas . . . . .	45
get_siope_despesas_funcao . . . . .	46
get_siope_indicadores . . . . .	48
get_siope_info_complementares . . . . .	50
get_siope_receitas . . . . .	51
get_siope_remuneracao . . . . .	53
get_siope_responsaveis . . . . .	54
get_siorg_estrutura . . . . .	56
get_siorg_orgaos . . . . .	57
get_siorg_unidade . . . . .	59

get_tc_estados . . . . .	60
get_tc_municipios . . . . .	61
get_tc_por_estados . . . . .	62
get_tc_por_estados_detalhe . . . . .	64
get_tc_por_municipio . . . . .	65
get_tc_por_municipio_detalhe . . . . .	67
get_tc_transferencias . . . . .	69
rreo_layout . . . . .	70
rreo_normalize_columns . . . . .	70
tesouror_clear_cache . . . . .	71
tidy_rreo . . . . .	72
<b>Index</b>	<b>74</b>

---

get_anexos	<i>Get report appendix reference table</i>
------------	--

---

## Description

Retrieves the reference table of report appendices (anexos) grouped by government sphere. This is a support table that describes which appendices are available for each report type (RREO, RGF, DCA, etc.).

## Usage

```
get_anexos(use_cache = TRUE, verbose = FALSE, page_size = NULL, max_rows = Inf)
```

```
get_annexes(use_cache = TRUE, verbose = FALSE,
            page_size = NULL, max_rows = Inf)
```

## Arguments

use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).

## Details

`get_annexes()` is an English alias for `get_anexos()`.

**Value**

A [tibble](#) with columns:

**esfera** Government sphere: "U" (Union), "E" (States), "M" (Municipalities).

**demonstrativo** Report type (e.g., "RREO", "RGF", "DCA").

**anexo** Appendix name (e.g., "RREO-Anexo 01").

**See Also**

Other SICONFI: [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

**Examples**

```
## Not run:
anexos <- get_anexos()

## End(Not run)
```

---

get_custos_demais	<i>Get other cost data</i>
-------------------	----------------------------

---

**Description**

Retrieves other (non-personnel) cost data for federal government organizations. All parameters are optional filters.

**Usage**

```
get_custos_demais(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_other(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.
natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.
organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_orgaos()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a mes filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to ano.
month	Integer. Month of the record (1-12). Optional. Maps to mes.
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to natureza_juridica.
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizations()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to organizacao_n1.
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n2.
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n3.

**Details**

`get_costs_other()` is an English alias.

**Value**

A [tibble](#) with other cost data.

## See Also

Other CUSTOS: [get\\_custos\\_depreciacao\(\)](#), [get\\_custos\\_pensionistas\(\)](#), [get\\_custos\\_pessoal\\_ativo\(\)](#), [get\\_custos\\_pessoal\\_inativo\(\)](#), [get\\_custos\\_transferencias\(\)](#)

## Examples

```
## Not run:
demaiss <- get_custos_demaiss(
  ano = 2023, mes = 6,
  organizacao_n1 = 244, # MEC
  organizacao_n2 = 249 # INEP
)

## End(Not run)
```

---

get\_custos\_depreciacao

*Get depreciation cost data*

---

## Description

Retrieves depreciation cost data for federal government organizations. All parameters are optional filters.

## Usage

```
get_custos_depreciacao(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_depreciation(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.
natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.
organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_orgaos()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a <code>mes</code> filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to <code>ano</code> .
month	Integer. Month of the record (1-12). Optional. Maps to <code>mes</code> .
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to <code>natureza_juridica</code> .
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizations()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to <code>organizacao_n1</code> .
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n2</code> .
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n3</code> .

**Details**

`get_costs_depreciation()` is an English alias.

**Value**

A [tibble](#) with depreciation cost data.

**See Also**

Other CUSTOS: [get\\_custos\\_demais\(\)](#), [get\\_custos\\_pensionistas\(\)](#), [get\\_custos\\_pessoal\\_ativo\(\)](#), [get\\_custos\\_pessoal\\_inativo\(\)](#), [get\\_custos\\_transferencias\(\)](#)

**Examples**

```
## Not run:
deprec <- get_custos_depreciacao(
  ano = 2023, mes = 6,
  organizacao_n1 = 244, # MEC
  organizacao_n2 = 249 # INEP
)

## End(Not run)
```

---

get\_custos\_pensionistas

*Get pensioner cost data*

---

**Description**

Retrieves cost data for federal government pensioners. All parameters are optional filters.

**Usage**

```
get_custos_pensionistas(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_pensioners(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.

natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.
organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizacao()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a <code>mes</code> filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to <code>ano</code> .
month	Integer. Month of the record (1-12). Optional. Maps to <code>mes</code> .
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to <code>natureza_juridica</code> .
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizacao()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to <code>organizacao_n1</code> .
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n2</code> .
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n3</code> .

### Details

`get_costs_pensioners()` is an English alias.

### Value

A [tibble](#) with pensioner cost data.

### See Also

Other CUSTOS: [get\\_custos\\_demais\(\)](#), [get\\_custos\\_depreciacao\(\)](#), [get\\_custos\\_pessoal\\_ativo\(\)](#), [get\\_custos\\_pessoal\\_inativo\(\)](#), [get\\_custos\\_transferencias\(\)](#)

**Examples**

```
## Not run:
pensionistas <- get_custos_pensionistas(
  ano = 2023, mes = 12,
  organizacao_n1 = 244, # MEC
  organizacao_n2 = 249 # INEP
)

## End(Not run)
```

---

```
get_custos_pessoal_ativo
  Get active staff cost data
```

---

**Description**

Retrieves cost data for active federal government staff. All parameters are optional filters.

**Usage**

```
get_custos_pessoal_ativo(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_active_staff(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.
natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.

organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_orgaos()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a <code>mes</code> filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to <code>ano</code> .
month	Integer. Month of the record (1-12). Optional. Maps to <code>mes</code> .
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to <code>natureza_juridica</code> .
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizations()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to <code>organizacao_n1</code> .
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n2</code> .
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to <code>organizacao_n3</code> .

### Details

**Performance:** The CUSTOS API is slow and unfiltered queries return hundreds of thousands of rows. Always filter by organization level (`organizacao_n1 + organizacao_n2`) and/or `mes` to get manageable results. Use `max_rows` for quick tests.

`get_costs_active_staff()` is an English alias.

### Value

A [tibble](#) with active staff cost data.

### See Also

Other CUSTOS: [get\\_custos\\_demais\(\)](#), [get\\_custos\\_depreciacao\(\)](#), [get\\_custos\\_pensionistas\(\)](#), [get\\_custos\\_pessoal\\_inativo\(\)](#), [get\\_custos\\_transferencias\(\)](#)

**Examples**

```
## Not run:
# WARNING: unfiltered queries return hundreds of thousands of rows
# and are very slow. Always filter by organization level and/or
# natureza_juridica + mes.

# Quick test: limit rows
ativos <- get_custos_pessoal_ativo(ano = 2023, mes = 6, max_rows = 100)

# Filter by ministry (N1) + entity (N2) for manageable results
# Use get_siorg_orgaos() to find SIORG codes
ativos_inep <- get_custos_pessoal_ativo(
  ano = 2023,
  organizacao_n1 = 244, # MEC (auto-padded to "000244")
  organizacao_n2 = 249 # INEP (auto-padded to "000249")
)

## End(Not run)
```

---

```
get_custos_pessoal_inativo
      Get retired staff cost data
```

---

**Description**

Retrieves cost data for retired (inactive) federal government staff. All parameters are optional filters.

**Usage**

```
get_custos_pessoal_inativo(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_retired_staff(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.
natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.
organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_orgaos()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a mes filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to ano.
month	Integer. Month of the record (1-12). Optional. Maps to mes.
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to natureza_juridica.
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizations()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to organizacao_n1.
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n2.
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n3.

**Details**

`get_costs_retired_staff()` is an English alias.

**Value**

A [tibble](#) with retired staff cost data.

**See Also**

Other CUSTOS: [get\\_custos\\_demais\(\)](#), [get\\_custos\\_depreciacao\(\)](#), [get\\_custos\\_pensionistas\(\)](#), [get\\_custos\\_pessoal\\_ativo\(\)](#), [get\\_custos\\_transferencias\(\)](#)

**Examples**

```
## Not run:
# Always filter to avoid slow, large queries
inativos <- get_custos_pessoal_inativo(
  ano = 2023, mes = 6,
  organizacao_n1 = 244, # MEC
  organizacao_n2 = 249 # INEP
)

## End(Not run)
```

---

get\_custos\_transferencias  
*Get transfer cost data*

---

**Description**

Retrieves transfer cost data for federal government organizations. All parameters are optional filters.

**Usage**

```
get_custos_transferencias(
  ano = NULL,
  mes = NULL,
  natureza_juridica = NULL,
  organizacao_n1 = NULL,
  organizacao_n2 = NULL,
  organizacao_n3 = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 500L,
  max_rows = Inf
)

get_costs_transfers(year = NULL, month = NULL,
  legal_nature = NULL, org_level1 = NULL, org_level2 = NULL,
  org_level3 = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = 500, max_rows = Inf)
```

**Arguments**

ano	Integer. Year of the record. Optional.
mes	Integer. Month of the record (1-12). Optional.
natureza_juridica	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional.
organizacao_n1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_orgaos()</a> to look up codes. You can pass plain codes (e.g., 244) — they are automatically zero-padded to the 6-digit format the API expects ("000244"). Optional.
organizacao_n2	Character or integer. SIORG code for the second-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
organizacao_n3	Character or integer. SIORG code for the third-level organization. Use <a href="#">get_siorg_estrutura()</a> to browse sub-units. Automatically zero-padded. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to FALSE.
page_size	Integer. Number of rows per API page. Defaults to 500. The CUSTOS backend is slow on broad queries: smaller pages are more robust against HTTP 504 time-outs at the cost of a few extra round-trips. The server default of 250 is conservative; 1000+ may time out. If the package returns <code>attr(result, "partial") = TRUE</code> , lower this further or add a mes filter.
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
year	Integer. Year of the record. Optional. Maps to ano.
month	Integer. Month of the record (1-12). Optional. Maps to mes.
legal_nature	Integer. Legal nature of the organization: 1 (Public Company), 2 (Public Foundation), 3 (Direct Administration), 4 (Autarchy), 6 (Mixed Economy Company). Optional. Maps to natureza_juridica.
org_level1	Character or integer. SIORG code for the top-level organization (Ministry level). Use <a href="#">get_siorg_organizations()</a> to look up codes. Plain codes (e.g., 244) are auto-padded to "000244". Optional. Maps to organizacao_n1.
org_level2	Character or integer. SIORG code (second level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n2.
org_level3	Character or integer. SIORG code (third level). Use <a href="#">get_siorg_structure()</a> to browse sub-units. Auto-padded. Optional. Maps to organizacao_n3.

**Details**

`get_costs_transfers()` is an English alias.

**Value**

A [tibble](#) with transfer cost data.

**See Also**

Other CUSTOS: [get\\_custos\\_demais\(\)](#), [get\\_custos\\_depreciacao\(\)](#), [get\\_custos\\_pensionistas\(\)](#), [get\\_custos\\_pessoal\\_ativo\(\)](#), [get\\_custos\\_pessoal\\_inativo\(\)](#)

**Examples**

```
## Not run:
transf <- get_custos_transferencias(
  ano = 2023, mes = 6,
  organizacao_n1 = 244, # MEC
  organizacao_n2 = 249 # INEP
)

## End(Not run)
```

---

get\_dca

*Get annual accounts data (DCA)*


---

**Description**

Retrieves data from the Annual Accounts Declaration (DCA) or the legacy QDCC for a specific entity and fiscal year.

**Usage**

```
get_dca(
  an_exercicio,
  id_ente,
  no_anexo = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_annual_accounts(fiscal_year, entity_id, appendix = NULL,
  use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

an_exercicio	Integer. Fiscal year (e.g., 2022). <b>Required.</b>
id_ente	Integer. IBGE code of the entity. <b>Required.</b>
no_anexo	Character. Appendix name filter (e.g., "DCA-Anexo I-AB"). Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .

page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., max_rows = 100).
fiscal_year	Integer. Fiscal year (e.g., 2022). <b>Required.</b> Maps to an_exercicio.
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to id_ente.
appendix	Character. Appendix name filter (e.g., "DCA-Anexo I-AB"). Optional. Maps to no_anexo.

### Details

get\_annual\_accounts() is an English alias for get\_dca().

### Value

A *tibble* with DCA/QDCC data including columns such as exercicio, instituicao, cod\_ibge, uf, anexo, rotulo, coluna, cod\_conta, conta, valor, and populacao.

### See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

### Examples

```
## Not run:
dca <- get_dca(an_exercicio = 2022, id_ente = 17)

## End(Not run)
```

---

get\_dca\_for\_state      *Get DCA data for all municipalities of a Brazilian state*

---

### Description

Fetches annual accounts (DCA) for every municipality of state\_uf, looping over [get\\_dca\(\)](#) with fault tolerance. See [get\\_rreo\\_for\\_state\(\)](#) for the rationale and behaviour of on\_error.

### Usage

```
get_dca_for_state(
  state_uf,
  an_exercicio,
  no_anexo = NULL,
  include_capital = TRUE,
  on_error = c("warn", "stop", "silent"),
```

```

    use_cache = TRUE,
    verbose = FALSE,
    page_size = NULL,
    max_rows = Inf
  )

```

```

get_annual_accounts_for_state(state_uf, fiscal_year, appendix = NULL,
  include_capital = TRUE, on_error = c("warn", "stop", "silent"),
  use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)

```

### Arguments

state_uf	Character. Two-letter UF code (e.g., "PE"). <b>Required.</b>
an_exercicio	Integer. Fiscal year. <b>Required.</b>
no_anexo	Character. Appendix name filter (e.g., "DCA-Anexo I-AB"). Optional.
include_capital	Logical. Include the state capital? Defaults to TRUE.
on_error	Character. "warn" (default), "stop", or "silent".
use_cache	Logical.
verbose	Logical.
page_size	Integer or NULL.
max_rows	Numeric.
fiscal_year	Integer. Fiscal year. <b>Required.</b> Maps to an_exercicio.
appendix	Character. Appendix name filter. Optional. Maps to no_anexo.

### Details

get\_annual\_accounts\_for\_state() is an English-parameter alias.

### Value

A [tibble](#) with all successful DCA rows. If any call failed, has an attribute "failed".

### See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

### Examples

```

## Not run:
dca_pe <- get_dca_for_state(state_uf = "PE", an_exercicio = 2022)

## End(Not run)

```

---

get_entes	<i>Get list of Brazilian government entities</i>
-----------	--

---

### Description

Retrieves the complete list of government entities (entes) registered in the SICONFI system, including states, municipalities, and the Federal District.

### Usage

```
get_entes(use_cache = TRUE, verbose = FALSE, page_size = NULL, max_rows = Inf)
```

```
get_entities(use_cache = TRUE, verbose = FALSE,
             page_size = NULL, max_rows = Inf)
```

### Arguments

use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to <code>Inf</code> (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).

### Details

`get_entities()` is an English alias for `get_entes()`.

### Value

A [tibble](#) with columns:

**cod\_ibge** IBGE code of the entity.

**ente** Name of the entity.

**capital** Whether the municipality is a state capital (1 = yes, 0 = no).

**regiao** Geographic region ("SU", "NE", "NO", "SE", "CO", "BR").

**uf** State abbreviation.

**esfera** Government sphere: "M", "E", "U", "D".

**an\_exercicio** Year of the population data.

**populacao** Estimated population.

**co\_cnpj** CNPJ of the entity.

**See Also**

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

**Examples**

```
## Not run:
entes <- get_entes()

## End(Not run)
```

---

get_extrato	<i>Get delivery status extract</i>
-------------	------------------------------------

---

**Description**

Retrieves the extract of report deliveries for a given entity and reference year. Useful for checking which reports have been submitted and their status (approved, rectified, etc.).

**Usage**

```
get_extrato(
  id_ente,
  an_referencia,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_delivery_status(entity_id, year, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

id_ente	Integer. IBGE code of the entity. <b>Required.</b>
an_referencia	Integer. Reference year (e.g., 2022). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to <code>id_ente</code> .
year	Integer. Reference year (e.g., 2022). <b>Required.</b> Maps to <code>an_referencia</code> .

## Details

get\_delivery\_status() is an English alias for get\_extrato().

## Value

A [tibble](#) with delivery status data including columns such as exercicio, cod\_ibge, instituicao, entregavel, periodo, periodicidade, status\_relatorio, data\_status, forma\_envio, and tipo\_relatorio.

## See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

## Examples

```
## Not run:
extrato <- get_extrato(id_ente = 17, an_referencia = 2022)

## End(Not run)
```

---

get_msc_controle	<i>Get MSC control accounts data</i>
------------------	--------------------------------------

---

## Description

Retrieves control accounts data (classes 7 and 8) from the Accounting Balances Matrix (MSC).

## Usage

```
get_msc_controle(
  id_ente,
  an_referencia,
  me_referencia,
  co_tipo_matriz,
  classe_conta,
  id_tv,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_msc_control(entity_id, year, month, matrix_type, account_class,
  value_type, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

id_ente	Integer. IBGE code of the entity. <b>Required.</b>
an_referencia	Integer. Reference year. <b>Required.</b>
me_referencia	Integer. Reference month (1-12). <b>Required.</b>
co_tipo_matriz	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b>
classe_conta	Integer. Account class: 7 or 8. <b>Required.</b>
id_tv	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to <code>id_ente</code> .
year	Integer. Reference year. <b>Required.</b> Maps to <code>an_referencia</code> .
month	Integer. Reference month (1-12). <b>Required.</b> Maps to <code>me_referencia</code> .
matrix_type	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b> Maps to <code>co_tipo_matriz</code> .
account_class	Integer. Account class: 7 or 8. <b>Required.</b> Maps to <code>classe_conta</code> .
value_type	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b> Maps to <code>id_tv</code> .

**Details**

`get_msc_control()` is an English alias for `get_msc_controle()`.

**Value**

A [tibble](#) with MSC control account data.

**See Also**

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

**Examples**

```
## Not run:
msc_ctrl <- get_msc_controle(
  id_ente = 17, an_referencia = 2022, me_referencia = 12,
  co_tipo_matriz = "MSCC", classe_conta = 8,
  id_tv = "ending_balance"
)

## End(Not run)
```

---

```
get_msc_orcamentaria Get MSC budgetary accounts data
```

---

**Description**

Retrieves budgetary accounts data (classes 5 and 6) from the Accounting Balances Matrix (MSC).

**Usage**

```
get_msc_orcamentaria(
  id_ente,
  an_referencia,
  me_referencia,
  co_tipo_matriz,
  classe_conta,
  id_tv,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_msc_budget(entity_id, year, month, matrix_type, account_class,
  value_type, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

id_ente	Integer. IBGE code of the entity. <b>Required.</b>
an_referencia	Integer. Reference year. <b>Required.</b>
me_referencia	Integer. Reference month (1-12). <b>Required.</b>
co_tipo_matriz	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b>
classe_conta	Integer. Account class: 5 or 6. <b>Required.</b>
id_tv	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b>

use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to <code>id_ente</code> .
year	Integer. Reference year. <b>Required.</b> Maps to <code>an_referencia</code> .
month	Integer. Reference month (1-12). <b>Required.</b> Maps to <code>me_referencia</code> .
matrix_type	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b> Maps to <code>co_tipo_matriz</code> .
account_class	Integer. Account class: 5 or 6. <b>Required.</b> Maps to <code>classe_conta</code> .
value_type	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b> Maps to <code>id_tv</code> .

## Details

`get_msc_budget()` is an English alias for `get_msc_orcamentaria()`.

## Value

A [tibble](#) with MSC budgetary account data.

## See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

## Examples

```
## Not run:
msc_orc <- get_msc_orcamentaria(
  id_ente = 17, an_referencia = 2022, me_referencia = 12,
  co_tipo_matriz = "MSCC", classe_conta = 6,
  id_tv = "period_change"
)

## End(Not run)
```

---

get\_msc\_patrimonial    *Get MSC equity/asset accounts data*

---

### Description

Retrieves equity and asset accounts data (classes 1 to 4) from the Accounting Balances Matrix (MSC).

### Usage

```
get_msc_patrimonial(
    id_ente,
    an_referencia,
    me_referencia,
    co_tipo_matriz,
    classe_conta,
    id_tv,
    use_cache = TRUE,
    verbose = FALSE,
    page_size = NULL,
    max_rows = Inf
)
```

```
get_msc_equity(entity_id, year, month, matrix_type, account_class,
    value_type, use_cache = TRUE, verbose = FALSE,
    page_size = NULL, max_rows = Inf)
```

### Arguments

id_ente	Integer. IBGE code of the entity. <b>Required.</b>
an_referencia	Integer. Reference year. <b>Required.</b>
me_referencia	Integer. Reference month (1-12). <b>Required.</b>
co_tipo_matriz	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b>
classe_conta	Integer. Account class: 1, 2, 3, or 4. <b>Required.</b>
id_tv	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).

entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to id_ente.
year	Integer. Reference year. <b>Required.</b> Maps to an_referencia.
month	Integer. Reference month (1-12). <b>Required.</b> Maps to me_referencia.
matrix_type	Character. Matrix type: "MSCC" (monthly aggregate) or "MSCE" (annual closing). <b>Required.</b> Maps to co_tipo_matriz.
account_class	Integer. Account class: 1, 2, 3, or 4. <b>Required.</b> Maps to classe_conta.
value_type	Character. Value type: "beginning_balance", "ending_balance", or "period_change". <b>Required.</b> Maps to id_tv.

### Details

get\_msc\_equity() is an English alias for get\_msc\_patrimonial().

### Value

A [tibble](#) with MSC equity/asset account data.

### See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

### Examples

```
## Not run:
msc_pat <- get_msc_patrimonial(
  id_ente = 17, an_referencia = 2022, me_referencia = 12,
  co_tipo_matriz = "MSCC", classe_conta = 1,
  id_tv = "ending_balance"
)

## End(Not run)
```

---

get\_opc\_cronograma\_liberacoes

*Get credit operation release schedule*

---

### Description

Retrieves the release schedule for credit operations linked to a PVL request. The id\_pleito can be obtained from [get\\_pvl\(\)](#).

**Usage**

```
get_opc_cronograma_liberacoes(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_credit_release_schedule(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

id_pleito	Integer. Database ID from <a href="#">get_pvl()</a> . <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
request_id	Integer. Database ID of the PVL request. Obtain from the <code>id_pleito</code> column of <a href="#">get_pvl()</a> results. <b>Required.</b> Maps to <code>id_pleito</code> .

**Details**

`get_credit_release_schedule()` is an English alias.

**Value**

A [tibble](#) with release schedule data.

**See Also**

Other SADIPEM: [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cdp\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

**Examples**

```
## Not run:
pvl_pe <- get_pvl(uf = "PE")
sched <- get_opc_cronograma_liberacoes(id_pleito = pvl_pe$id_pleito[1])

## End(Not run)
```

---

get\_opc\_cronograma\_pagamentos

*Get credit operation payment schedule*

---

### Description

Retrieves the payment schedule for credit operations linked to a PVL request. The `id_pleito` can be obtained from [get\\_pvl\(\)](#).

### Usage

```
get_opc_cronograma_pagamentos(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)
```

```
get_credit_payment_schedule(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

### Arguments

<code>id_pleito</code>	Integer. Database ID from <a href="#">get_pvl()</a> . <b>Required.</b>
<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.
<code>verbose</code>	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
<code>page_size</code>	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
<code>max_rows</code>	Numeric. Maximum number of rows to return. Defaults to <code>Inf</code> (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
<code>request_id</code>	Integer. Database ID of the PVL request. Obtain from the <code>id_pleito</code> column of <a href="#">get_pvl()</a> results. <b>Required.</b> Maps to <code>id_pleito</code> .

### Details

`get_credit_payment_schedule()` is an English alias.

### Value

A [tibble](#) with payment schedule data.

### See Also

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cdp\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

**Examples**

```
## Not run:
pvl_pe <- get_pvl(uf = "PE")
pgto <- get_opc_cronograma_pagamentos(id_pleito = pvl_pe$id_pleito[1])

## End(Not run)
```

---

get\_opc\_taxa\_cambio    *Get credit operation exchange rate data*

---

**Description**

Retrieves exchange rate data for credit operations linked to a PVL request. The `id_pleito` can be obtained from [get\\_pvl\(\)](#).

**Usage**

```
get_opc_taxa_cambio(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_credit_exchange_rate(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

<code>id_pleito</code>	Integer. Database ID from <a href="#">get_pvl()</a> . <b>Required.</b>
<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.
<code>verbose</code>	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
<code>page_size</code>	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
<code>max_rows</code>	Numeric. Maximum number of rows to return. Defaults to <code>Inf</code> (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
<code>request_id</code>	Integer. Database ID of the PVL request. Obtain from the <code>id_pleito</code> column of <a href="#">get_pvl()</a> results. <b>Required.</b> Maps to <code>id_pleito</code> .

**Details**

`get_credit_exchange_rate()` is an English alias.

**Value**

A [tibble](#) with exchange rate data.

**See Also**

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_pvl\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cdp\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

**Examples**

```
## Not run:
cambio <- get_opc_taxa_cambio(id_pleito = 40353)

## End(Not run)
```

---

get\_pvl

*Get public debt verification requests (PVL)*

---

**Description**

Retrieves data from the Public Debt Verification Letters (PVL) system. These are requests for approval of credit operations and debt by subnational entities. Use the resulting `id_plei` to column to query detail functions like [get\\_pvl\\_tramitacao\(\)](#), [get\\_opc\\_cronograma\\_liberacoes\(\)](#), etc.

**Usage**

```
get_pvl(
  uf = NULL,
  tipo_interessado = NULL,
  id_ente = NULL,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_debt_requests(state = NULL, entity_type = NULL,
  entity_id = NULL, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

<code>uf</code>	Character. State abbreviation (e.g., "PE"). Optional.
<code>tipo_interessado</code>	Character. Type of requesting entity (e.g., "Município", "Estado"). Optional.
<code>id_ente</code>	Integer. IBGE code of the requesting entity. Optional.
<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.

verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
state	Character. State abbreviation (e.g., "PE"). Optional. Maps to <code>uf</code> .
entity_type	Character. Type of requesting entity (e.g., "Município", "Estado"). Optional. Maps to <code>tipo_interessado</code> .
entity_id	Integer. IBGE code of the requesting entity. Optional. Maps to <code>id_ente</code> .

### Details

`get_debt_requests()` is an English alias.

### Value

A [tibble](#) with PVL data including `id_pleito`, `tipo_interessado`, `interessado`, `cod_ibge`, `uf`, `status`, `tipo_operacao`, `finalidade`, `credor`, `valor`, and more.

### See Also

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cdp\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

### Examples

```
## Not run:
# Search all PVLs for Pernambuco
pvl_pe <- get_pvl(uf = "PE")

# Search by IBGE code (Recife)
pvl_recife <- get_pvl(id_ente = 2611606)

# Use id_pleito from results for detail queries:
id <- pvl_pe$id_pleito[1]
pagamentos <- get_opc_cronograma_pagamentos(id_pleito = id)
cdp <- get_res_cdp(id_pleito = id)

# For PVL processing status, filter approved PVLs first:
deferidos <- pvl_pe[pvl_pe$status == "Deferido", ]
if (nrow(deferidos) > 0) {
  status <- get_pvl_tramitacao(id_pleito = deferidos$id_pleito[1])
}

## End(Not run)
```

---

get\_pvl\_tramitacao      *Get PVL processing status (approved non-credit operations)*

---

### Description

Retrieves processing status for PVL requests that were **approved (deferred)** and involve **non-credit operations**. This endpoint only returns data for PVLs with status "Deferido". For most PVLs (e.g., archived, in progress, or credit operations), this will return an empty tibble.

### Usage

```
get_pvl_tramitacao(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)
```

```
get_pvl_status(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

### Arguments

id_pleito	Integer. Database ID of a <b>deferred</b> PVL request (from the id_pleito column of <code>get_pvl()</code> ). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
request_id	Integer. Database ID of a <b>deferred</b> PVL request. Obtain from the id_pleito column of <code>get_pvl()</code> results, filtered by <code>status == "Deferido"</code> . <b>Required.</b> Maps to id_pleito.

### Details

To find PVLs that have data in this endpoint, filter by `status == "Deferido"` in the results of `get_pvl()`.

`get_pvl_status()` is an English alias.

### Value

A **tibble** with PVL processing status data. Returns an empty tibble if the PVL is not an approved non-credit operation.

**See Also**

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\(\)](#), [get\\_res\\_cdp\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

**Examples**

```
## Not run:
# Step 1: search PVLs and filter for approved ones
pvl_pe <- get_pvl(uf = "PE")
pvl_deferidos <- pvl_pe[pvl_pe$status == "Deferido", ]

# Step 2: pick an id_pleito from the APPROVED requests
id <- pvl_deferidos$id_pleito[1]

# Step 3: get processing status (only works for deferred PVLs)
status <- get_pvl_tramitacao(id_pleito = id)

## End(Not run)
```

---

get\_res\_cdp

*Get debt capacity result (CDP)*


---

**Description**

Retrieves the Debt Capacity Result (Resultado da Capacidade de Pagamento) linked to a PVL request. The `id_pleito` can be obtained from [get\\_pvl\(\)](#).

**Usage**

```
get_res_cdp(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_debt_capacity(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

<code>id_pleito</code>	Integer. Database ID from <a href="#">get_pvl()</a> . <b>Required.</b>
<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.
<code>verbose</code>	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .

page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., max_rows = 100).
request_id	Integer. Database ID of the PVL request. Obtain from the id_pleito column of <a href="#">get_pvl()</a> results. <b>Required.</b> Maps to id_pleito.

### Details

`get_debt_capacity()` is an English alias.

### Value

A [tibble](#) with CDP result data.

### See Also

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cronograma\\_pagamentos\(\)](#)

### Examples

```
## Not run:
cdp <- get_res_cdp(id_pleito = 40353)

## End(Not run)
```

---

```
get_res_cronograma_pagamentos
  Get debt payment schedule result
```

---

### Description

Retrieves the payment schedule result linked to a PVL request. The `id_pleito` can be obtained from [get\\_pvl\(\)](#).

### Usage

```
get_res_cronograma_pagamentos(
  id_pleito,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_debt_payment_schedule(request_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

id_pleito	Integer. Database ID from <a href="#">get_pvl()</a> . <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
request_id	Integer. Database ID of the PVL request. Obtain from the <code>id_pleito</code> column of <a href="#">get_pvl()</a> results. <b>Required.</b> Maps to <code>id_pleito</code> .

**Details**

`get_debt_payment_schedule()` is an English alias.

**Value**

A [tibble](#) with payment schedule result data.

**See Also**

Other SADIPEM: [get\\_opc\\_cronograma\\_liberacoes\(\)](#), [get\\_opc\\_cronograma\\_pagamentos\(\)](#), [get\\_opc\\_taxa\\_cambio\(\)](#), [get\\_pvl\(\)](#), [get\\_pvl\\_tramitacao\(\)](#), [get\\_res\\_cdp\(\)](#)

**Examples**

```
## Not run:
pgto_res <- get_res_cronograma_pagamentos(id_pleito = 40353)

## End(Not run)
```

---

get\_rgf

*Get Fiscal Management Report data (RGF)*

---

**Description**

Retrieves data from the Fiscal Management Report (RGF) for specific filtering criteria. The RGF contains information about personnel expenses, debt, credit operations, and other fiscal indicators.

**Usage**

```

get_rgf(
  an_exercicio,
  in_periodicidade,
  nr_periodo,
  co_tipo_demonstrativo,
  no_anexo,
  co_esfera,
  co_poder,
  id_ente,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

```

```

get_fiscal_report(fiscal_year, periodicity, period, report_type,
  appendix, sphere, branch, entity_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)

```

**Arguments**

an_exercicio	Integer. Fiscal year (e.g., 2022). <b>Required.</b>
in_periodicidade	Character. Periodicity: "Q" (four-monthly) or "S" (semi-annual). <b>Required.</b>
nr_periodo	Integer. Period number (1-3 for four-monthly, 1-2 for semi-annual). <b>Required.</b>
co_tipo_demonstrativo	Character. Report type: "RGF" or "RGF Simplificado". <b>Required.</b>
no_anexo	Character. Appendix name (e.g., "RGF-Anexo 01"). <b>Required.</b>
co_esfera	Character. Government sphere: "M" (municipalities), "E" (states), or "U" (union). <b>Required.</b>
co_poder	Character. Government branch: "E" (executive), "L" (legislative), "J" (judiciary), "M" (public ministry), "D" (public defender). <b>Required.</b>
id_ente	Integer. IBGE code of the entity. <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
fiscal_year	Integer. Fiscal year (e.g., 2022). <b>Required.</b> Maps to <code>an_exercicio</code> .
periodicity	Character. Periodicity: "Q" (quadrimester) or "S" (semester). <b>Required.</b> Maps to <code>in_periodicidade</code> .

period	Integer. Period number: 1-3 (quadrimester) or 1-2 (semester). <b>Required.</b> Maps to nr_periodo.
report_type	Character. Report type: "RGF" or "RGF Simplificado". <b>Required.</b> Maps to co_tipo_demonstrativo.
appendix	Character. Appendix name (e.g., "RGF-Anexo 01"). <b>Required.</b> Maps to no_anexo.
sphere	Character. Government sphere: "M", "E", or "U". <b>Required.</b> Maps to co_esfera.
branch	Character. Government branch: "E" (executive), "L" (legislative), "J" (judiciary), "M" (public ministry), "D" (public defender). <b>Required.</b> Maps to co_poder.
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to id_ente.

### Details

get\_fiscal\_report() is an English-parameter alias for this function.

### Value

A [tibble](#) with RGF data.

### See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

### Examples

```
## Not run:
rgf <- get_rgf(
  an_exercicio = 2022, in_periodicidade = "Q", nr_periodo = 3,
  co_tipo_demonstrativo = "RGF", no_anexo = "RGF-Anexo 01",
  co_esfera = "E", co_poder = "E", id_ente = 17
)

## End(Not run)
```

---

get\_rgf\_for\_state      *Get RGF data for all municipalities of a Brazilian state*

---

### Description

Fetches RGF data for every municipality of state\_uf, looping over [get\\_rgf\(\)](#) with fault tolerance. See [get\\_rreo\\_for\\_state\(\)](#) for the rationale and behaviour of on\_error.

**Usage**

```

get_rgf_for_state(
    state_uf,
    an_exercicio,
    in_periodicidade,
    nr_perodo,
    co_tipo_demonstrativo,
    no_anexo,
    co_poder,
    include_capital = TRUE,
    on_error = c("warn", "stop", "silent"),
    use_cache = TRUE,
    verbose = FALSE,
    page_size = NULL,
    max_rows = Inf
)

```

```

get_fiscal_report_for_state(state_uf, fiscal_year, periodicity,
    period, report_type, appendix, branch,
    include_capital = TRUE, on_error = c("warn", "stop", "silent"),
    use_cache = TRUE, verbose = FALSE,
    page_size = NULL, max_rows = Inf)

```

**Arguments**

state_uf	Character. Two-letter UF code (e.g., "PE"). <b>Required.</b>
an_exercicio	Integer. Fiscal year. <b>Required.</b>
in_periodicidade	Character. "Q" (four-monthly) or "S" (semi-annual). <b>Required.</b>
nr_perodo	Integer. Period number. <b>Required.</b>
co_tipo_demonstrativo	Character. "RGF" or "RGF Simplificado". <b>Required.</b>
no_anexo	Character. Appendix name (e.g., "RGF-Anexo 01"). <b>Required.</b>
co_poder	Character. Government branch: "E", "L", "J", "M", "D". <b>Required.</b>
include_capital	Logical. Include the state capital? Defaults to TRUE.
on_error	Character. "warn" (default), "stop", or "silent".
use_cache	Logical.
verbose	Logical.
page_size	Integer or NULL.
max_rows	Numeric.
fiscal_year	Integer. Fiscal year. <b>Required.</b> Maps to an_exercicio.
periodicity	Character. "Q" or "S". <b>Required.</b> Maps to in_periodicidade.
period	Integer. Period number. <b>Required.</b> Maps to nr_perodo.

report_type	Character. "RGF" or "RGF Simplificado". <b>Required.</b> Maps to co_tipo_demonstrativo.
appendix	Character. Appendix name. <b>Required.</b> Maps to no_anexo.
branch	Character. Government branch. <b>Required.</b> Maps to co_poder.

### Details

get\_fiscal\_report\_for\_state() is an English-parameter alias.

### Value

A [tibble](#) with all successful RGF rows. If any call failed, has an attribute "failed".

### See Also

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rreo\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

### Examples

```
## Not run:
rgf_pe <- get_rgf_for_state(
  state_uf = "PE", an_exercicio = 2022,
  in_periodicidade = "Q", nr_periodo = 3,
  co_tipo_demonstrativo = "RGF", no_anexo = "RGF-Anexo 01",
  co_poder = "E"
)

## End(Not run)
```

---

get\_rreo

*Get Budget Execution Summary Report data (RREO)*

---

### Description

Retrieves data from the Budget Execution Summary Report (RREO) for specific filtering criteria. The RREO is published bimonthly and contains information about revenues, expenses, and other budgetary data.

### Usage

```
get_rreo(
  an_exercicio,
  nr_periodo,
  co_tipo_demonstrativo,
  no_anexo,
  co_esfera,
  id_ente,
```

```

    use_cache = TRUE,
    verbose = FALSE,
    page_size = NULL,
    max_rows = Inf
  )

```

```

get_budget_report(fiscal_year, period, report_type, appendix,
  sphere, entity_id, use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)

```

## Arguments

an_exercicio	Integer. Fiscal year (e.g., 2022). <b>Required.</b>
nr_periodo	Integer. Bimester number (1-6). <b>Required.</b>
co_tipo_demonstrativo	Character. Report type: "RREO" or "RREO Simplificado". <b>Required.</b> Note: "RREO Simplificado" applies only to municipalities with fewer than 50,000 inhabitants that opted for simplified reporting.
no_anexo	Character. Appendix name (e.g., "RREO-Anexo 01"). <b>Required.</b>
co_esfera	Character. Government sphere: "M" (municipalities), "E" (states), or "U" (union). <b>Required.</b>
id_ente	Integer. IBGE code of the entity. <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
page_size	Integer or NULL. Number of rows per API page. If NULL (default), uses the API server default (5000 for SICONFI/SADIPEM).
max_rows	Numeric. Maximum number of rows to return. Defaults to Inf (all rows). Useful for quick tests with large datasets (e.g., <code>max_rows = 100</code> ).
fiscal_year	Integer. Fiscal year (e.g., 2022). <b>Required.</b> Maps to <code>an_exercicio</code> .
period	Integer. Bimester number (1-6). <b>Required.</b> Maps to <code>nr_periodo</code> .
report_type	Character. Report type: "RREO" or "RREO Simplificado". <b>Required.</b> Maps to <code>co_tipo_demonstrativo</code> .
appendix	Character. Appendix name (e.g., "RREO-Anexo 01"). <b>Required.</b> Maps to <code>no_anexo</code> .
sphere	Character. Government sphere: "M" (municipalities), "E" (states), or "U" (union). <b>Required.</b> Maps to <code>co_esfera</code> .
entity_id	Integer. IBGE code of the entity. <b>Required.</b> Maps to <code>id_ente</code> .

## Details

`get_budget_report()` is an English-parameter alias for this function.

## Value

A [tibble](#) with RREO data including columns such as `exercicio`, `demonstrativo`, `periodo`, `periodicidade`, `instituicao`, `cod_ibge`, `uf`, `populacao`, `anexo`, `rotulo`, `coluna`, `cod_conta`, `conta`, and `valor`.

**See Also**

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\\_for\\_state\(\)](#)

**Examples**

```
## Not run:
rreo <- get_rreo(
  an_exercicio = 2022, nr_periodo = 6,
  co_tipo_demonstrativo = "RREO",
  no_anexo = "RREO-Anexo 01",
  co_esfera = "E", id_ente = 17
)

## End(Not run)
```

---

get\_rreo\_for\_state      *Get RREO data for all municipalities of a Brazilian state*

---

**Description**

Fetches RREO data for every municipality of state\_uf, looping over [get\\_rreo\(\)](#) with fault tolerance: if an individual municipality call fails after all retries, the failure is recorded and the loop continues. Failed calls are returned in attr(result, "failed").

**Usage**

```
get_rreo_for_state(
  state_uf,
  an_exercicio,
  nr_periodo,
  co_tipo_demonstrativo,
  no_anexo,
  include_capital = TRUE,
  on_error = c("warn", "stop", "silent"),
  use_cache = TRUE,
  verbose = FALSE,
  page_size = NULL,
  max_rows = Inf
)

get_budget_report_for_state(state_uf, fiscal_year, period,
  report_type, appendix, include_capital = TRUE,
  on_error = c("warn", "stop", "silent"),
  use_cache = TRUE, verbose = FALSE,
  page_size = NULL, max_rows = Inf)
```

**Arguments**

state_uf	Character. Two-letter UF code (e.g., "PE", "ES"). <b>Required.</b>
an_exercicio	Integer. Fiscal year. <b>Required.</b>
nr_perodo	Integer. Bimester (1-6). <b>Required.</b>
co_tipo_demonstrativo	Character. "RREO" or "RREO Simplificado". <b>Required.</b>
no_anexo	Character. Appendix name (e.g., "RREO-Anexo 01"). <b>Required.</b>
include_capital	Logical. Include the state capital? Defaults to TRUE.
on_error	Character. One of "warn" (default — log and continue), "stop" (abort on first failure), or "silent" (record but no message).
use_cache	Logical. If TRUE (default), uses the in-memory cache.
verbose	Logical. If TRUE, prints the full API URL for each call.
page_size	Integer or NULL. Rows per API page.
max_rows	Numeric. Maximum rows per municipality call.
fiscal_year	Integer. Fiscal year. <b>Required.</b> Maps to an_exercicio.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to nr_perodo.
report_type	Character. "RREO" or "RREO Simplificado". <b>Required.</b> Maps to co_tipo_demonstrativo.
appendix	Character. Appendix name. <b>Required.</b> Maps to no_anexo.

**Details**

This is the recommended way to assemble a state-wide panel: it handles pagination per municipality, uses the cache, and surfaces partial failures instead of aborting on the first error (see SICONFI behaviour for entities that have not yet homologated a given report).

get\_budget\_report\_for\_state() is an English-parameter alias.

**Value**

A [tibble](#) with RREO rows for all successful municipalities. If any call failed, has an attribute "failed" (tibble with iteration, id, error).

**See Also**

Other SICONFI: [get\\_anexos\(\)](#), [get\\_dca\(\)](#), [get\\_dca\\_for\\_state\(\)](#), [get\\_entes\(\)](#), [get\\_extrato\(\)](#), [get\\_msc\\_controle\(\)](#), [get\\_msc\\_orcamentaria\(\)](#), [get\\_msc\\_patrimonial\(\)](#), [get\\_rgf\(\)](#), [get\\_rgf\\_for\\_state\(\)](#), [get\\_rreo\(\)](#)

**Examples**

```
## Not run:
rreo_es <- get_rreo_for_state(
  state_uf = "ES", an_exercicio = 2021, nr_perodo = 6,
  co_tipo_demonstrativo = "RREO", no_anexo = "RREO-Anexo 01"
)
```

```
attr(rreo_es, "failed")

## End(Not run)
```

---

```
get_siope_dados_gerais
```

*Get SIOPE general data*

---

### Description

Retrieves general data from SIOPE including demographics, GDP, revenues, expenses, and declaration metadata for municipalities and states.

### Usage

```
get_siope_dados_gerais(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
  orderby = NULL,
  select = NULL
)

get_siope_general_data(year, period, state, use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)
```

### Arguments

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.

filter	Character. OData \$filter expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a max_rows = 1 query and use toupper(names(result)). Optional.
orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., c("NOM_MUNI", "VAL_DECL")). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year (e.g., 2023). <b>Required.</b> Maps to ano.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to periodo.
state	Character. State abbreviation (e.g., "PE"). <b>Required.</b> Maps to uf.

### Details

**Performance tip:** use the filter parameter to narrow results on the server before downloading. For example, filtering by municipality name or IBGE code returns only the rows you need instead of all municipalities in the state.

get\_siope\_general\_data() is an English alias.

### Value

A [tibble](#) with 52 columns including tipo, num\_ano, sig\_uf, cod\_muni, nom\_muni, num\_popu, revenue and expense values, PIB data, and declaration metadata.

### See Also

Other SIOPE: [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis](#)

### Examples

```
## Not run:
# General data for all municipalities in Pernambuco
dados <- get_siope_dados_gerais(ano = 2023, periodo = 6, uf = "PE")

# FAST: Filter to a single municipality (server-side)
recife <- get_siope_dados_gerais(
  ano = 2023, periodo = 6, uf = "PE",
  filter = "NOM_MUNI eq 'Recife'"
)

# Filter + select specific columns (use original API names!)
resumo <- get_siope_dados_gerais(
  ano = 2023, periodo = 6, uf = "PE",
  filter = "COD_MUNI eq 2611606",
  select = c("NOM_MUNI", "VAL_RECE_REAL", "VAL_DESP_PAGA")
)
```

```
)
## End(Not run)
```

---

```
get_siope_despesas      Get SIOPE education expenses data
```

---

## Description

Retrieves education spending data broken down by expense categories for municipalities and states.

## Usage

```
get_siope_despesas(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
  orderby = NULL,
  select = NULL
)

get_siope_expenses(year, period, state, use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)
```

## Arguments

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData <code>\$filter</code> expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.

orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., c("NOM_MUNI", "VAL_DECL")). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required.</b> Maps to ano.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to periodo.
state	Character. State abbreviation. <b>Required.</b> Maps to uf.

### Details

get\_siope\_expenses() is an English alias.

### Value

A [tibble](#) with 21 columns including expense categories and values.

### See Also

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis\(\)](#)

### Examples

```
## Not run:
desp <- get_siope_despesas(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

```
get_siope_despesas_funcao
```

*Get SIOPE expenses by education function*

---

### Description

Retrieves education expenses broken down by government function (e.g., basic education, higher education).

### Usage

```
get_siope_despesas_funcao(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
```

```

    max_rows = Inf,
    filter = NULL,
    orderby = NULL,
    select = NULL
  )

```

```

get_siope_expenses_by_function(year, period, state,
  use_cache = TRUE, verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)

```

### Arguments

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData <code>\$filter</code> expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.
orderby	Character. OData <code>\$orderby</code> expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., <code>c("NOM_MUNI", "VAL_DECL")</code> ). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required.</b> Maps to ano.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to periodo.
state	Character. State abbreviation. <b>Required.</b> Maps to uf.

### Details

`get_siope_expenses_by_function()` is an English alias.

### Value

A [tibble](#) with 12 columns including function codes and expense values.

**See Also**

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis](#)

**Examples**

```
## Not run:
desp_func <- get_siope_despesas_funcao(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

get\_siope\_indicadores *Get SIOPE education indicators*

---

**Description**

Retrieves education spending indicators such as percentage of revenue applied to education, FUNDEB compliance, etc.

**Usage**

```
get_siope_indicadores(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
  orderby = NULL,
  select = NULL
)

get_siope_indicators(year, period, state, use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)
```

**Arguments**

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.

verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData \$filter expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.
orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., <code>c("NOM_MUNI", "VAL_DECL")</code> ). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required</b> . Maps to ano.
period	Integer. Bimester (1-6). <b>Required</b> . Maps to periodo.
state	Character. State abbreviation. <b>Required</b> . Maps to uf.

## Details

`get_siope_indicadores()` is an English alias.

## Value

A [tibble](#) with 13 columns including indicator codes and values.

## See Also

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis](#)

## Examples

```
## Not run:
ind <- get_siope_indicadores(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

```
get_siope_info_complementares
```

*Get SIOPE supplementary information*

---

## Description

Retrieves supplementary information from SIOPE declarations.

## Usage

```
get_siope_info_complementares(
    ano,
    periodo,
    uf,
    use_cache = TRUE,
    verbose = FALSE,
    page_size = 1000L,
    max_rows = Inf,
    filter = NULL,
    orderby = NULL,
    select = NULL
)
```

```
get_siope_supplementary(year, period, state, use_cache = TRUE,
    verbose = FALSE,
    page_size = 1000, max_rows = Inf,
    filter = NULL, orderby = NULL, select = NULL)
```

## Arguments

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData <code>\$filter</code> expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.

orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., c("NOM_MUNI", "VAL_DECL")). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required.</b> Maps to ano.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to periodo.
state	Character. State abbreviation. <b>Required.</b> Maps to uf.

### Details

get\_siope\_supplementary() is an English alias.

### Value

A [tibble](#) with 10 columns.

### See Also

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis\(\)](#)

### Examples

```
## Not run:
info <- get_siope_info_complementares(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

get\_siope\_receitas      *Get SIOPE education revenue data*

---

### Description

Retrieves education revenue data including tax revenues, transfers, and FUNDEB contributions.

### Usage

```
get_siope_receitas(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
```

```

    orderby = NULL,
    select = NULL
  )

get_siope_revenues(year, period, state, use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)

```

### Arguments

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData \$filter expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.
orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., c("NOM_MUNI", "VAL_DECL")). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required.</b> Maps to ano.
period	Integer. Bimester (1-6). <b>Required.</b> Maps to periodo.
state	Character. State abbreviation. <b>Required.</b> Maps to uf.

### Details

`get_siope_revenues()` is an English alias.

### Value

A [tibble](#) with 14 columns including revenue categories and values.

### See Also

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_remuneracao\(\)](#), [get\\_siope\\_responsaveis\(\)](#)

**Examples**

```
## Not run:
rec <- get_siope_receitas(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

get\_siope\_remuneracao *Get SIOPE staff compensation data*

---

**Description**

Retrieves compensation data for education professionals. This endpoint has an additional required parameter: mes / month for the specific month within the year.

**Usage**

```
get_siope_remuneracao(
  ano,
  periodo,
  mes,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
  orderby = NULL,
  select = NULL
)
```

```
get_siope_compensation(year, period, month, state,
  use_cache = TRUE, verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)
```

**Arguments**

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
mes	Integer. Month of the fiscal year (1-12). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL.
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.

filter	Character. OData \$filter expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with and/or. Column names must use the <b>original API names</b> (uppercase), not the snake_case cleaned names. To discover valid names, run a max_rows = 1 query and use toupper(names(result)). Optional.
orderby	Character. OData \$orderby expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., c("NOM_MUNI", "VAL_DECL")). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required</b> . Maps to ano.
period	Integer. Bimester (1-6). <b>Required</b> . Maps to periodo.
month	Integer. Month (1-12). <b>Required</b> . Maps to mes.
state	Character. State abbreviation. <b>Required</b> . Maps to uf.

### Details

get\_siope\_compensation() is an English alias.

### Value

A [tibble](#) with 19 columns including compensation categories and values for education professionals.

### See Also

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_responsaveis](#)

### Examples

```
## Not run:
rem <- get_siope_remuneracao(ano = 2023, periodo = 6, mes = 12, uf = "PE")

## End(Not run)
```

---

get\_siope\_responsaveis

*Get SIOPE officials/responsible persons data*

---

### Description

Retrieves data about the officials responsible for education spending declarations, including mayors, secretaries, and accountants.

**Usage**

```
get_siope_responsaveis(
  ano,
  periodo,
  uf,
  use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000L,
  max_rows = Inf,
  filter = NULL,
  orderby = NULL,
  select = NULL
)
```

```
get_siope_officials(year, period, state, use_cache = TRUE,
  verbose = FALSE,
  page_size = 1000, max_rows = Inf,
  filter = NULL, orderby = NULL, select = NULL)
```

**Arguments**

ano	Integer. Year of the data (e.g., 2023). <b>Required.</b>
periodo	Integer. Bimester period (1-6). <b>Required.</b>
uf	Character. State abbreviation (e.g., "PE"). <b>Required.</b>
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to FALSE. Set globally with <code>options(tesouror.verbose = TRUE)</code> .
page_size	Integer. Rows per page for OData pagination. Defaults to 1000.
max_rows	Numeric. Maximum rows to return. Defaults to Inf.
filter	Character. OData <code>\$filter</code> expression to narrow results <b>on the server</b> (much faster than downloading everything). Uses OData syntax (e.g., "NOM_MUNI eq 'Recife'", "COD_MUNI eq 2611606"). Combine with <code>and/or</code> . Column names must use the <b>original API names</b> (uppercase), not the <code>snake_case</code> cleaned names. To discover valid names, run a <code>max_rows = 1</code> query and use <code>toupper(names(result))</code> . Optional.
orderby	Character. OData <code>\$orderby</code> expression to sort results (e.g., "NOM_MUNI asc", "NUM_POPU desc"). Uses original API column names (uppercase). Optional.
select	Character vector. Column names to return (reduces payload size). Uses original API column names (e.g., <code>c("NOM_MUNI", "VAL_DECL")</code> ). If a column name is invalid the API returns HTTP 400. Optional.
year	Integer. Year. <b>Required.</b> Maps to <code>ano</code> .
period	Integer. Bimester (1-6). <b>Required.</b> Maps to <code>periodo</code> .
state	Character. State abbreviation. <b>Required.</b> Maps to <code>uf</code> .

**Details**

get\_siope\_officials() is an English alias.

**Value**

A [tibble](#) with 35 columns including official names, CPFs, roles, and contact information.

**See Also**

Other SIOPE: [get\\_siope\\_dados\\_gerais\(\)](#), [get\\_siope\\_despesas\(\)](#), [get\\_siope\\_despesas\\_funcao\(\)](#), [get\\_siope\\_indicadores\(\)](#), [get\\_siope\\_info\\_complementares\(\)](#), [get\\_siope\\_receitas\(\)](#), [get\\_siope\\_remuneracao\(\)](#)

**Examples**

```
## Not run:
resp <- get_siope_responsaveis(ano = 2023, periodo = 6, uf = "PE")

## End(Not run)
```

---

get\_siorg\_estrutura    *Get organizational structure from SIORG*

---

**Description**

Retrieves the organizational structure tree for a specific SIORG unit, returning all sub-units (departments, secretariats, etc.) as a flat tibble.

**Usage**

```
get_siorg_estrutura(
  codigo_unidade,
  vinculados = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_siorg_structure(unit_code, include_linked = NULL,
  use_cache = TRUE, verbose = FALSE)
```

**Arguments**

codigo_unidade	Integer or character. SIORG code of the unit. <b>Required.</b> Use <a href="#">get_siorg_orgaos()</a> to find codes.
vinculados	Character. Return linked organs/entities? "SIM" or "NAO". Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .

unit\_code Integer or character. SIORG code. Maps to codigo\_unidade.  
 include\_linked Character. "SIM" or "NAO". Maps to vinculados.

### Details

Use the returned codes as organizacao\_n2 / org\_level2 and organizacao\_n3 / org\_level3 in CUSTOS API functions.

get\_siorg\_structure() is an English alias.

### Value

A [tibble](#) with columns including codigo\_unidade, codigo\_unidade\_pai, nome, sigla, codigo\_tipo\_unidade. Use codigo\_unidade\_pai to navigate the parent-child hierarchy.

### See Also

Other SIORG: [get\\_siorg\\_orgaos\(\)](#), [get\\_siorg\\_unidade\(\)](#)

### Examples

```
## Not run:
# Get structure of AGU (code 46)
estrutura <- get_siorg_estrutura(codigo_unidade = 46)

## End(Not run)
```

---

get_siorg_orgaos	<i>Get federal organizations from SIORG</i>
------------------	---

---

### Description

Retrieves the list of federal government organizations (organs and entities) from the SIORG system. Use the returned SIORG codes as organizacao\_n1 / org\_level1 in CUSTOS API functions like [get\\_custos\\_pessoal\\_ativo\(\)](#).

### Usage

```
get_siorg_orgaos(
  codigo_poder = NULL,
  codigo_esfera = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_siorg_organizations(power_code = NULL, sphere_code = NULL,
  use_cache = TRUE, verbose = FALSE)
```

**Arguments**

codigo_poder	Integer. Power branch code: 1 = Executive, 2 = Legislative, 3 = Judiciary. Optional (default returns all).
codigo_esfera	Integer. Government sphere: 1 = Federal, 2 = State/District, 3 = Municipal. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Defaults to <code>getOption("tesouror.verbose")</code> , FALSE).
power_code	Integer. Power branch: 1 = Executive, 2 = Legislative, 3 = Judiciary. Maps to <code>codigo_poder</code> .
sphere_code	Integer. Sphere: 1 = Federal. Maps to <code>codigo_esfera</code> .

**Details**

`get_siorg_organizations()` is an English alias.

**Value**

A [tibble](#) with columns including `codigo_unidade` (SIORG code — use as `organizacao_n1` in CUSTOS), `codigo_unidade_pai` (parent code — for navigating hierarchy), `nome`, `sigla`, `codigo_tipo_unidade` ("orgao" or "entidade"), `codigo_natureza_juridica`, `codigo_esfera`, `codigo_poder`.

**See Also**

Other SIORG: [get\\_siorg\\_estrutura\(\)](#), [get\\_siorg\\_unidade\(\)](#)

**Examples**

```
## Not run:
# List all federal executive branch organizations
orgaos <- get_siorg_orgaos(codigo_poder = 1, codigo_esfera = 1)

# Find a specific ministry by name
mec <- orgaos[grepl("Educação", orgaos$nome), ]
mec_code <- mec$codigo_unidade[mec$sigla == "MEC"]

# Use that code in CUSTOS queries (auto-padded to "000244")
custos <- get_custos_pessoal_ativo(
  ano = 2023, organizacao_n1 = mec_code
)

# Browse children: units whose parent is MEC
mec_filhos <- orgaos[orgaos$codigo_unidade_pai == mec_code, ]

## End(Not run)
```

---

get\_siorg\_unidade      *Get details of a single SIORG unit*

---

### Description

Retrieves summary data for a single organizational unit by its SIORG code. Returns a single-row tibble with the unit's details.

### Usage

```
get_siorg_unidade(codigo_unidade, use_cache = TRUE, verbose = FALSE)
```

```
get_siorg_unit(unit_code, use_cache = TRUE, verbose = FALSE)
```

### Arguments

codigo\_unidade Integer or character. SIORG code. **Required.**

use\_cache Logical. If TRUE (default), uses an in-memory cache.

verbose Logical. If TRUE, prints the full API URL. Defaults to `getOption("tesouror.verbose", FALSE)`.

unit\_code Integer or character. SIORG code. Maps to `codigo_unidade`.

### Details

`get_siorg_unit()` is an English alias.

### Value

A [tibble](#) (single row) with unit details including `codigo_unidade`, `nome`, `sigla`, `codigo_tipo_unidade`, `codigo_unidade_pai`, `codigo_natureza_juridica`, and more.

### See Also

Other SIORG: [get\\_siorg\\_estrutura\(\)](#), [get\\_siorg\\_orgaos\(\)](#)

### Examples

```
## Not run:  
# Get details for AGU (code 46)  
agu <- get_siorg_unidade(codigo_unidade = 46)  
  
## End(Not run)
```

---

get\_tc\_estados            *Get state dictionary*

---

### Description

Retrieves the dictionary of states and their numeric codes. Use these codes in the `p_estado / state_code` parameter of other functions.

### Usage

```
get_tc_estados(use_cache = TRUE, verbose = FALSE)
```

```
get_tc_states(use_cache = TRUE, verbose = FALSE)
```

### Arguments

`use_cache`            Logical. If TRUE (default), uses an in-memory cache.

`verbose`             Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to `getOption("tesouror.verbose", FALSE)`.

### Details

These are **internal Treasury codes**, not IBGE codes.

`get_tc_states()` is an English alias.

### Value

A [tibble](#) with columns `codigo` (numeric code) and `nome` (name) of each state.

### See Also

Other Transferencias: [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_estados\\_detalhe\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_por\\_municipio\\_detalhe\(\)](#), [get\\_tc\\_transferencias\(\)](#)

Other Transferencias dictionaries: [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_transferencias\(\)](#)

### Examples

```
## Not run:
estados <- get_tc_estados()
# Use estados$codigo as values for p_estado in other functions

## End(Not run)
```

---

get\_tc\_municipios      *Get municipality dictionary*

---

### Description

Retrieves the dictionary of municipalities and their numeric codes. Use these codes in the `p_municipio` / `municipality` parameter of other functions.

### Usage

```
get_tc_municipios(  
  p_nome = NULL,  
  p_uf = NULL,  
  use_cache = TRUE,  
  verbose = FALSE  
)
```

```
get_tc_municipalities(name = NULL, state_code = NULL,  
  use_cache = TRUE, verbose = FALSE)
```

### Arguments

<code>p_nome</code>	Character. Partial municipality name for searching (e.g., "Recife"). Optional.
<code>p_uf</code>	Numeric or character. State code from <a href="#">get_tc_estados()</a> . <b>Not</b> a state abbreviation or IBGE code. Optional.
<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.
<code>verbose</code>	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
<code>name</code>	Character. Partial municipality name for searching (e.g., "Recife"). Optional. Maps to <code>p_nome</code> .
<code>state_code</code>	Numeric or character. State code from <a href="#">get_tc_estados()</a> . <b>Not</b> a state abbreviation or IBGE code. Optional. Maps to <code>p_uf</code> .

### Details

These are **internal Treasury codes**, not IBGE codes. Filter by state using the numeric code from [get\\_tc\\_estados\(\)](#).

`get_tc_municipalities()` is an English alias.

### Value

A [tibble](#) with municipality data including `codigo`, `codigo_uf`, and `nome`.

**See Also**

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_estados\\_detalhe\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_por\\_municipio\\_detalhe\(\)](#), [get\\_tc\\_transferencias\(\)](#)

Other Transferencias dictionaries: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_transferencias\(\)](#)

**Examples**

```
## Not run:
# Step 1: find the state code (Treasury code, NOT IBGE)
estados <- get_tc_estados()
pe_code <- estados$codigo[estados$nome == "Pernambuco"]

# Step 2: list municipalities for that state
municipios <- get_tc_municipios(p_uf = pe_code)

# Or search by partial name
recife <- get_tc_municipios(p_nome = "Recife")

## End(Not run)
```

---

get\_tc\_por\_estados      *Get constitutional transfers by state*

---

**Description**

Retrieves constitutional transfer data aggregated by state.

**Usage**

```
get_tc_por_estados(
  p_estado = NULL,
  p_ano = NULL,
  p_mes = NULL,
  p_transferencia = NULL,
  p_sn_detalhar = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_tc_by_state(state_code = NULL, year = NULL, month = NULL,
  transfer_type = NULL, detailed = NULL, use_cache = TRUE, verbose = FALSE)
```

**Arguments**

**p\_estado**            State code(s) from [get\\_tc\\_estados\(\)](#). Accepts a vector or a colon-separated string. Optional.

**p\_ano**                Year(s). Accepts a vector or colon-separated string. Optional.

p_mes	Month(s) (1-12). Accepts a vector or colon-separated string. Optional.
p_transferencia	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional.
p_sn_detalhar	Character. Set to any value to include detailed breakdown. Omit for summary. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
state_code	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector (e.g., <code>c(1, 2)</code> ) or colon-separated string ("1:2"). These are <b>Treasury codes</b> , not IBGE codes. Optional. Maps to p_estado.
year	Year(s). Accepts a vector or colon-separated string. Optional. Maps to p_ano.
month	Month(s) (1-12). Accepts a vector or colon-separated string. Optional. Maps to p_mes.
transfer_type	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional. Maps to p_transferencia.
detailed	Character. Set to any value to include a detailed breakdown. Omit for summary. Optional. Maps to p_sn_detalhar.

### Details

All codes are **internal Treasury codes** (not IBGE). Use the dictionary functions to look them up: [get\\_tc\\_estados\(\)](#) for state codes and [get\\_tc\\_transferencias\(\)](#) for transfer type codes.

Multi-value parameters accept either a colon-separated string ("1:2:3") or an R vector (`c(1, 2, 3)`).

`get_tc_by_state()` is an English alias.

### Value

A [tibble](#) with transfer data by state.

### See Also

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\\_detalhe\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_por\\_municipio\\_detalhe\(\)](#), [get\\_tc\\_transferencias\(\)](#)

### Examples

```
## Not run:
# Step 1: look up codes (Treasury codes, NOT IBGE)
estados <- get_tc_estados()
tipos <- get_tc_transferencias()

pe_code <- estados$codigo[estados$nome == "Pernambuco"]

# Step 2: query (pass a vector or colon-separated string)
```

```
tc_pe <- get_tc_por_estados(p_estado = pe_code, p_ano = 2023)
tc_multi <- get_tc_por_estados(
  p_estado = c(1, 2), p_ano = 2023, p_mes = c(1, 2)
)

## End(Not run)
```

---

```
get_tc_por_estados_detalhe
```

*Get detailed constitutional transfers by state*

---

## Description

Retrieves detailed constitutional transfer data by state, with full breakdown of each transfer.

## Usage

```
get_tc_por_estados_detalhe(
  p_estado = NULL,
  p_ano = NULL,
  p_mes = NULL,
  p_transferencia = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_tc_by_state_detail(state_code = NULL, year = NULL,
  month = NULL, transfer_type = NULL, use_cache = TRUE, verbose = FALSE)
```

## Arguments

p_estado	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. Optional.
p_ano	Year(s). Accepts a vector or colon-separated string. Optional.
p_mes	Month(s). Accepts a vector or colon-separated string. Optional.
p_transferencia	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
state_code	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. <b>Treasury codes</b> , not IBGE. Optional. Maps to p_estado.
year	Year(s). Accepts a vector or colon-separated string. Optional. Maps to p_ano.
month	Month(s) (1-12). Accepts a vector or colon-separated string. Optional. Maps to p_mes.

transfer\_type Transfer type code(s) from [get\\_tc\\_transferencias\(\)](#). Accepts a vector or colon-separated string. Optional. Maps to p\_transferencia.

### Details

All codes are **internal Treasury codes** (not IBGE). See [get\\_tc\\_estados\(\)](#) and [get\\_tc\\_transferencias\(\)](#) for dictionaries.

[get\\_tc\\_by\\_state\\_detail\(\)](#) is an English alias.

### Value

A [tibble](#) with detailed transfer data by state.

### See Also

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_por\\_municipio\\_detalhe\(\)](#), [get\\_tc\\_transferencias\(\)](#)

### Examples

```
## Not run:
estados <- get_tc_estados()
pe_code <- estados$codigo[estados$nome == "Pernambuco"]
detalhe <- get_tc_por_estados_detalhe(p_estado = pe_code, p_ano = 2023)

## End(Not run)
```

---

get\_tc\_por\_municipio *Get constitutional transfers by municipality*

---

### Description

Retrieves constitutional transfer data for municipalities within states.

### Usage

```
get_tc_por_municipio(
  p_estado = NULL,
  p_municipio = NULL,
  p_ano = NULL,
  p_mes = NULL,
  p_transferencia = NULL,
  p_sn_detalhar = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_tc_by_municipality(state_code = NULL, municipality = NULL,
  year = NULL, month = NULL, transfer_type = NULL, detailed = NULL,
  use_cache = TRUE, verbose = FALSE)
```

**Arguments**

p_estado	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. Optional.
p_municipio	Municipality code(s) from <a href="#">get_tc_municipios()</a> . Accepts a vector or colon-separated string. Optional.
p_ano	Year(s). Accepts a vector or colon-separated string. Optional.
p_mes	Month(s). Accepts a vector or colon-separated string. Optional.
p_transferencia	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional.
p_sn_detalhar	Character. Set to any value to include detailed breakdown. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
state_code	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. <b>Treasury codes</b> , not IBGE. Optional. Maps to p_estado.
municipality	Municipality code(s) from <a href="#">get_tc_municipios()</a> . Accepts a vector or colon-separated string. <b>Treasury codes</b> . Optional. Maps to p_municipio.
year	Year(s). Accepts a vector or colon-separated string. Optional. Maps to p_ano.
month	Month(s) (1-12). Accepts a vector or colon-separated string. Optional. Maps to p_mes.
transfer_type	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional. Maps to p_transferencia.
detailed	Character. Set to any value to include detail. Optional. Maps to p_sn_detalhar.

**Details**

All codes are **internal Treasury codes** (not IBGE). Use the dictionary functions to look them up: [get\\_tc\\_estados\(\)](#) for state codes, [get\\_tc\\_municipios\(\)](#) for municipality codes, and [get\\_tc\\_transferencias\(\)](#) for transfer type codes.

Multi-value parameters accept either a colon-separated string ("1:2:3") or an R vector (c(1, 2, 3)).

[get\\_tc\\_by\\_municipality\(\)](#) is an English alias.

**Value**

A [tibble](#) with transfer data by municipality.

**See Also**

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_estados\\_detalhe\(\)](#), [get\\_tc\\_por\\_municipio\\_detalhe\(\)](#), [get\\_tc\\_transferencias\(\)](#)

**Examples**

```
## Not run:
# Step 1: look up codes (Treasury codes, NOT IBGE)
estados <- get_tc_estados()
pe_code <- estados$codigo[estados$nome == "Pernambuco"]
municipios <- get_tc_municipios(p_uf = pe_code)
recife_code <- municipios$codigo[municipios$nome == "Recife"]

# Step 2: query (pass vector or string)
tc <- get_tc_por_municipio(
  p_estado = pe_code,
  p_municipio = recife_code,
  p_ano = 2023
)

## End(Not run)
```

---

```
get_tc_por_municipio_detalhe
```

*Get detailed constitutional transfers by municipality*

---

**Description**

Retrieves detailed constitutional transfer data by municipality.

**Usage**

```
get_tc_por_municipio_detalhe(
  p_estado = NULL,
  p_municipio = NULL,
  p_ano = NULL,
  p_mes = NULL,
  p_transferencia = NULL,
  use_cache = TRUE,
  verbose = FALSE
)

get_tc_by_municipality_detail(state_code = NULL,
  municipality = NULL, year = NULL, month = NULL,
  transfer_type = NULL, use_cache = TRUE, verbose = FALSE)
```

**Arguments**

p_estado	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. Optional.
p_municipio	Municipality code(s) from <a href="#">get_tc_municipios()</a> . Accepts a vector or colon-separated string. Optional.

p_ano	Year(s). Accepts a vector or colon-separated string. Optional.
p_mes	Month(s). Accepts a vector or colon-separated string. Optional.
p_transferencia	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional.
use_cache	Logical. If TRUE (default), uses an in-memory cache.
verbose	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .
state_code	State code(s) from <a href="#">get_tc_estados()</a> . Accepts a vector or colon-separated string. <b>Treasury codes</b> , not IBGE. Optional. Maps to p_estado.
municipality	Municipality code(s) from <a href="#">get_tc_municipios()</a> . Accepts a vector or colon-separated string. <b>Treasury codes</b> . Optional. Maps to p_municipio.
year	Year(s). Accepts a vector or colon-separated string. Optional. Maps to p_ano.
month	Month(s) (1-12). Accepts a vector or colon-separated string. Optional. Maps to p_mes.
transfer_type	Transfer type code(s) from <a href="#">get_tc_transferencias()</a> . Accepts a vector or colon-separated string. Optional. Maps to p_transferencia.

## Details

All codes are **internal Treasury codes** (not IBGE). See [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), and [get\\_tc\\_transferencias\(\)](#) for dictionaries.

`get_tc_by_municipality_detail()` is an English alias.

## Value

A [tibble](#) with detailed municipality transfer data.

## See Also

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_estados\\_detalhe\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_transferencias\(\)](#)

## Examples

```
## Not run:
estados <- get_tc_estados()
pe_code <- estados$codigo[estados$nome == "Pernambuco"]
det <- get_tc_por_municipio_detalhe(p_estado = pe_code, p_ano = 2023)

## End(Not run)
```

---

get\_tc\_transferencias *Get transfer type dictionary*

---

### Description

Retrieves the dictionary of constitutional transfer types and their numeric codes. Use these codes in the `p_transferencia / transfer_type` parameter of other functions.

### Usage

```
get_tc_transferencias(use_cache = TRUE, verbose = FALSE)
```

```
get_tc_transfer_types(use_cache = TRUE, verbose = FALSE)
```

### Arguments

<code>use_cache</code>	Logical. If TRUE (default), uses an in-memory cache.
<code>verbose</code>	Logical. If TRUE, prints the full API URL being called. Useful for debugging or testing in a browser. Defaults to <code>getOption("tesouror.verbose", FALSE)</code> .

### Details

These are **internal Treasury codes**, not IBGE codes.

`get_tc_transfer_types()` is an English alias.

### Value

A [tibble](#) with columns `codigo` (code) and `nome` (name) of each transfer type.

### See Also

Other Transferencias: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#), [get\\_tc\\_por\\_estados\(\)](#), [get\\_tc\\_por\\_estados\\_detalle\(\)](#), [get\\_tc\\_por\\_municipio\(\)](#), [get\\_tc\\_por\\_municipio\\_detalle\(\)](#)

Other Transferencias dictionaries: [get\\_tc\\_estados\(\)](#), [get\\_tc\\_municipios\(\)](#)

### Examples

```
## Not run:  
tipos <- get_tc_transferencias()  
# Use tipos$codigo as values for p_transferencia in other functions  
  
## End(Not run)
```

---

rreo_layout	<i>Return the bundled RREO layout reference table</i>
-------------	---

---

### Description

RREO appendix names, account labels, and column suffixes drift across fiscal years (e.g., RREO-Anexo 04.3 - RGPS for 2019-2022 vs. RREO-Anexo 04.4 - RGPS for 2023+). The package ships a small reference table in `inst/extdata/rreo_layout.csv` that maps (`topic`, `regime`, `year_range`) to the correct appendix and account-matching key.

### Usage

```
rreo_layout()
```

### Details

`tidy_rreo()` uses this table to assemble layout-stable indicators across years; you can also use it directly to look up which appendix to fetch with `get_rreo()` for a given topic.

### Value

A *tibble* with columns `topic`, `regime`, `first_year`, `last_year`, `co_esfera`, `no_anexo`, `conta_match`, `indicador`.

### See Also

Other RREO tidy: `rreo_normalize_columns()`, `tidy_rreo()`

### Examples

```
rreo_layout()
```

---

rreo_normalize_columns	<i>Normalize the columna field of a RREO tibble across years</i>
------------------------	--

---

### Description

SICONFI's RREO column labels drift over years: the same conceptual column appears as "DESPEAS LIQUIDADAS ATÉ O BIMESTRE / 2019" in 2019, "DESPEAS LIQUIDADAS ATÉ O BIMESTRE" (no year) in 2021-2022, and "DESPEAS LIQUIDADAS ATÉ O BIMESTRE / 2023" in 2023+. This helper adds two columns:

### Usage

```
rreo_normalize_columns(data)
```

## Arguments

`data` A tibble returned by `get_rreo()` or `get_rreo_for_state()`. Must contain a `coluna` column.

## Details

- `coluna_padrao`: the column label with any trailing `"/ YYYY"` or `"EM YYYY"` suffix removed (whitespace squished).
- `coluna_ano`: the year that appeared in the suffix (integer), or NA when no year was present. Useful for distinguishing the current-year column from a comparative previous-year column.

## Value

The input tibble with `coluna_padrao` and `coluna_ano` appended.

## See Also

Other RREO tidy: `rreo_layout()`, `tidy_rreo()`

## Examples

```
demo <- tibble::tibble(  
  coluna = c(  
    "DESPEAS LIQUIDADAS ATÉ O BIMESTRE / 2023",  
    "DESPEAS LIQUIDADAS ATÉ O BIMESTRE",  
    "INSCRITAS EM RESTOS A PAGAR NÃO PROCESSADOS EM 2023"  
  )  
)  
rreo_normalize_columns(demo)
```

---

`tesouror_clear_cache` *Clear the tesouror in-memory cache*

---

## Description

Removes **all** cached API responses stored during the current R session. This applies to every API covered by the package: SICONFI, CUSTOS, SADIPEM, and Transferencias Constitucionais. All cached responses share the same in-memory store and are cleared together.

## Usage

```
tesouror_clear_cache()
```

## Value

Invisible NULL.

**Examples**

```
tesouror_clear_cache()
```

---

tidy\_rreo

*Tidy a RREO tibble by topic, reconciling layout drift across years*


---

**Description**

Filters a long RREO tibble (typically produced by [get\\_rreo\(\)](#)) down to the rows that match a known indicator for topic (and optionally regime), using the rules in [rreo\\_layout\(\)](#). Account labels are matched on a year-stable, accent-folded key (Roman numerals and formula text are stripped before comparison), so the same call returns a coherent series across years even when SICONFI re-labelled the appendix or account.

**Usage**

```
tidy_rreo(data, topic, regime = NULL)
```

**Arguments**

data	A tibble returned by <a href="#">get_rreo()</a> or <a href="#">get_rreo_for_state()</a> , with at least the columns <code>exercicio</code> , <code>conta</code> , <code>coluna</code> , <code>valor</code> .
topic	Character. Topic name (e.g., "previdencia"). See <a href="#">rreo_layout()</a> for the supported set.
regime	Optional character. Filter to a subset of regimes within topic (e.g., "rgps", "rpps"). When NULL (default) all regimes are kept.

**Details**

Currently supported topics:

- "previdencia" — federal previdência (RGPS, RPPS civis, FCDF, militares inativos) for the União sphere. Anexos 04.1 / 04.2 / 04.3 / 04.4 of the RREO; the layout knows that the RGPS appendix moved from 04.3 (up to 2022) to 04.4 (2023+) and that civis/FCDF moved from 04.1 (up to 2022) to 04.2 (2023+).

Pull requests adding new topics to `inst/extdata/rreo_layout.csv` are welcome.

**Value**

A [tibble](#) with the matched rows, plus extra columns:

- `indicador`: stable indicator name (e.g., "resultado\_previdenciario\_rgps").
- `regime`: matched regime.
- `coluna_padrao`, `coluna_ano`: see [rreo\\_normalize\\_columns\(\)](#).

**See Also**

Other RREO tidy: [rreo\\_layout\(\)](#), [rreo\\_normalize\\_columns\(\)](#)

**Examples**

```
## Not run:
library(dplyr)
rreo <- purrr::map_dfr(2019:2023, \(yr) {
  get_rreo(an_exercicio = yr, nr_perodo = 6,
          co_tipo_demonstrativo = "RREO",
          no_anexo = rreo_layout()$no_anexo[
            rreo_layout()$topic == "previdencia" &
            rreo_layout()$regime == "rgps" &
            yr >= rreo_layout()$first_year &
            yr <= rreo_layout()$last_year
          ][1],
          co_esfera = "U", id_ente = 1)
})
tidy_rreo(rreo, topic = "previdencia", regime = "rgps") |>
  filter(coluna_padrao == "DESPESAS LIQUIDADAS ATÉ O BIMESTRE",
         is.na(coluna_ano) | coluna_ano == exercicio) |>
  select(exercicio, indicador, valor)

## End(Not run)
```

# Index

- \* **CUSTOS**
  - get\_custos\_demais, 4
  - get\_custos\_depreciacao, 6
  - get\_custos\_pensionistas, 8
  - get\_custos\_pessoal\_ativo, 10
  - get\_custos\_pessoal\_inativo, 12
  - get\_custos\_transferencias, 14
- \* **RREO tidy**
  - rreo\_layout, 70
  - rreo\_normalize\_columns, 70
  - tidy\_rreo, 72
- \* **SADIPEM**
  - get\_opc\_cronograma\_liberacoes, 26
  - get\_opc\_cronograma\_pagamentos, 28
  - get\_opc\_taxa\_cambio, 29
  - get\_pvl, 30
  - get\_pvl\_tramitacao, 32
  - get\_res\_cdp, 33
  - get\_res\_cronograma\_pagamentos, 34
- \* **SICONFI**
  - get\_anexos, 3
  - get\_dca, 16
  - get\_dca\_for\_state, 17
  - get\_entes, 19
  - get\_extrato, 20
  - get\_msc\_controle, 21
  - get\_msc\_orcamentaria, 23
  - get\_msc\_patrimonial, 25
  - get\_rgf, 35
  - get\_rgf\_for\_state, 37
  - get\_rreo, 39
  - get\_rreo\_for\_state, 41
- \* **SIOPE**
  - get\_siope\_dados\_gerais, 43
  - get\_siope\_despesas, 45
  - get\_siope\_despesas\_funcao, 46
  - get\_siope\_indicadores, 48
  - get\_siope\_info\_complementares, 50
  - get\_siope\_receitas, 51
  - get\_siope\_remuneracao, 53
  - get\_siope\_responsaveis, 54
- \* **SIORG dictionaries**
  - get\_siorg\_orgaos, 57
- \* **SIORG**
  - get\_siorg\_estutura, 56
  - get\_siorg\_orgaos, 57
  - get\_siorg\_unidade, 59
- \* **Transferencias dictionaries**
  - get\_tc\_estados, 60
  - get\_tc\_municipios, 61
  - get\_tc\_transferencias, 69
- \* **Transferencias**
  - get\_tc\_estados, 60
  - get\_tc\_municipios, 61
  - get\_tc\_por\_estados, 62
  - get\_tc\_por\_estados\_detalhe, 64
  - get\_tc\_por\_municipio, 65
  - get\_tc\_por\_municipio\_detalhe, 67
  - get\_tc\_transferencias, 69
- get\_anexos, 3, 17, 18, 20–22, 24, 26, 37, 39, 41, 42
- get\_annexes (get\_anexos), 3
- get\_annual\_accounts (get\_dca), 16
- get\_annual\_accounts\_for\_state (get\_dca\_for\_state), 17
- get\_budget\_report (get\_rreo), 39
- get\_budget\_report\_for\_state (get\_rreo\_for\_state), 41
- get\_costs\_active\_staff (get\_custos\_pessoal\_ativo), 10
- get\_costs\_depreciation (get\_custos\_depreciacao), 6
- get\_costs\_other (get\_custos\_demais), 4
- get\_costs\_pensioners (get\_custos\_pensionistas), 8
- get\_costs\_retired\_staff (get\_custos\_pessoal\_inativo), 12

- get\_costs\_transfers
  - (get\_custos\_transferencias), 14
- get\_credit\_exchange\_rate
  - (get\_opc\_taxa\_cambio), 29
- get\_credit\_payment\_schedule
  - (get\_opc\_cronograma\_pagamentos), 28
- get\_credit\_release\_schedule
  - (get\_opc\_cronograma\_liberacoes), 26
- get\_custos\_demais, 4, 8, 9, 11, 14, 16
- get\_custos\_depreciacao, 6, 6, 9, 11, 14, 16
- get\_custos\_pensionistas, 6, 8, 8, 11, 14, 16
- get\_custos\_pessoal\_ativo, 6, 8, 9, 10, 14, 16
- get\_custos\_pessoal\_ativo(), 57
- get\_custos\_pessoal\_inativo, 6, 8, 9, 11, 12, 16
- get\_custos\_transferencias, 6, 8, 9, 11, 14, 14
- get\_dca, 4, 16, 18, 20–22, 24, 26, 37, 39, 41, 42
- get\_dca(), 17
- get\_dca\_for\_state, 4, 17, 17, 20–22, 24, 26, 37, 39, 41, 42
- get\_debt\_capacity (get\_res\_cdp), 33
- get\_debt\_payment\_schedule
  - (get\_res\_cronograma\_pagamentos), 34
- get\_debt\_requests (get\_pvl), 30
- get\_delivery\_status (get\_extrato), 20
- get\_entes, 4, 17, 18, 19, 21, 22, 24, 26, 37, 39, 41, 42
- get\_entities (get\_entes), 19
- get\_extrato, 4, 17, 18, 20, 20, 22, 24, 26, 37, 39, 41, 42
- get\_fiscal\_report (get\_rgf), 35
- get\_fiscal\_report\_for\_state
  - (get\_rgf\_for\_state), 37
- get\_msc\_budget (get\_msc\_orcamentaria), 23
- get\_msc\_control (get\_msc\_controle), 21
- get\_msc\_controle, 4, 17, 18, 20, 21, 21, 24, 26, 37, 39, 41, 42
- get\_msc\_equity (get\_msc\_patrimonial), 25
- get\_msc\_orcamentaria, 4, 17, 18, 20–22, 23, 26, 37, 39, 41, 42
- get\_msc\_patrimonial, 4, 17, 18, 20–22, 24, 25, 37, 39, 41, 42
- get\_opc\_cronograma\_liberacoes, 26, 28, 30, 31, 33–35
- get\_opc\_cronograma\_liberacoes(), 30
- get\_opc\_cronograma\_pagamentos, 27, 28, 30, 31, 33–35
- get\_opc\_taxa\_cambio, 27, 28, 29, 31, 33–35
- get\_pvl, 27, 28, 30, 30, 33–35
- get\_pvl(), 26–29, 32–35
- get\_pvl\_status (get\_pvl\_tramitacao), 32
- get\_pvl\_tramitacao, 27, 28, 30, 31, 32, 34, 35
- get\_pvl\_tramitacao(), 30
- get\_res\_cdp, 27, 28, 30, 31, 33, 33, 35
- get\_res\_cronograma\_pagamentos, 27, 28, 30, 31, 33, 34, 34
- get\_rgf, 4, 17, 18, 20–22, 24, 26, 35, 39, 41, 42
- get\_rgf(), 37
- get\_rgf\_for\_state, 4, 17, 18, 20–22, 24, 26, 37, 37, 41, 42
- get\_rreo, 4, 17, 18, 20–22, 24, 26, 37, 39, 39, 42
- get\_rreo(), 41, 70–72
- get\_rreo\_for\_state, 4, 17, 18, 20–22, 24, 26, 37, 39, 41, 41
- get\_rreo\_for\_state(), 17, 37, 71, 72
- get\_siope\_compensation
  - (get\_siope\_remuneracao), 53
- get\_siope\_dados\_gerais, 43, 46, 48, 49, 51, 52, 54, 56
- get\_siope\_despesas, 44, 45, 48, 49, 51, 52, 54, 56
- get\_siope\_despesas\_funcao, 44, 46, 46, 49, 51, 52, 54, 56
- get\_siope\_expenses
  - (get\_siope\_despesas), 45
- get\_siope\_expenses\_by\_function
  - (get\_siope\_despesas\_funcao), 46
- get\_siope\_general\_data
  - (get\_siope\_dados\_gerais), 43
- get\_siope\_indicadores, 44, 46, 48, 48, 51, 52, 54, 56
- get\_siope\_indicators
  - (get\_siope\_indicadores), 48
- get\_siope\_info\_complementares, 44, 46, 48, 49, 50, 52, 54, 56
- get\_siope\_officials

- (get\_siopre\_responsaveis), 54
- get\_siopre\_receitas, 44, 46, 48, 49, 51, 51, 54, 56
- get\_siopre\_remuneracao, 44, 46, 48, 49, 51, 52, 53, 56
- get\_siopre\_responsaveis, 44, 46, 48, 49, 51, 52, 54, 54
- get\_siopre\_revenues
  - (get\_siopre\_receitas), 51
- get\_siopre\_supplementary
  - (get\_siopre\_info\_complementares), 50
- get\_siorg\_estrutura, 56, 58, 59
- get\_siorg\_estrutura(), 5, 7, 9, 11, 13, 15
- get\_siorg\_organizations
  - (get\_siorg\_orgaos), 57
- get\_siorg\_organizations(), 5, 7, 9, 11, 13, 15
- get\_siorg\_orgaos, 57, 57, 59
- get\_siorg\_orgaos(), 5, 7, 9, 11, 13, 15, 56
- get\_siorg\_structure
  - (get\_siorg\_estrutura), 56
- get\_siorg\_structure(), 5, 7, 9, 11, 13, 15
- get\_siorg\_unidade, 57, 58, 59
- get\_siorg\_unit (get\_siorg\_unidade), 59
- get\_tc\_by\_municipality
  - (get\_tc\_por\_municipio), 65
- get\_tc\_by\_municipality\_detail
  - (get\_tc\_por\_municipio\_detalhe), 67
- get\_tc\_by\_state (get\_tc\_por\_estados), 62
- get\_tc\_by\_state\_detail
  - (get\_tc\_por\_estados\_detalhe), 64
- get\_tc\_estados, 60, 62, 63, 65, 66, 68, 69
- get\_tc\_estados(), 61–68
- get\_tc\_municipalities
  - (get\_tc\_municipios), 61
- get\_tc\_municipios, 60, 61, 63, 65, 66, 68, 69
- get\_tc\_municipios(), 66–68
- get\_tc\_por\_estados, 60, 62, 62, 65, 66, 68, 69
- get\_tc\_por\_estados\_detalhe, 60, 62, 63, 64, 66, 68, 69
- get\_tc\_por\_municipio, 60, 62, 63, 65, 65, 68, 69
- get\_tc\_por\_municipio\_detalhe, 60, 62, 63, 65, 66, 67, 69
- get\_tc\_states (get\_tc\_estados), 60
- get\_tc\_transfer\_types
  - (get\_tc\_transferencias), 69
- get\_tc\_transferencias, 60, 62, 63, 65, 66, 68, 69
- get\_tc\_transferencias(), 63–66, 68
- rreo\_layout, 70, 71, 73
- rreo\_layout(), 72
- rreo\_normalize\_columns, 70, 70, 73
- rreo\_normalize\_columns(), 72
- tesouror\_clear\_cache, 71
- tibble, 4, 5, 7, 9, 11, 13, 15, 17–19, 21, 22, 24, 26–28, 30–32, 34, 35, 37, 39, 40, 42, 44, 46, 47, 49, 51, 52, 54, 56–61, 63, 65, 66, 68–70, 72
- tidy\_rreo, 70, 71, 72
- tidy\_rreo(), 70