# Package 'ineptr2'

April 28, 2026

**Title** Wrapper for Statistics Portugal API

**Version** 0.3.0

**Description** An R6-based client to facilitate interaction with the Statistics Portugal (Instituto Nacional de Estatistica - INE) API (<https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_api&INST=322751522&xlang=en>).

**License** MIT + file LICENSE

**URL** https://c-matos.github.io/ineptr2/, https://github.com/c-matos/ineptr2

**BugReports** https://github.com/c-matos/ineptr2/issues

**Imports** httr2, jsonlite, R6, rlang, xml2

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

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Carlos Matos [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-1134-0396>)

**Maintainer** Carlos Matos <carlosmdmatos@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-04-28 20:10:02 UTC

## Contents

`INEClient` *INE API Client*

## Description

An R6 class providing access to the Statistics Portugal (INE) API. Holds configuration state (language, caching preferences) and provides methods for retrieving data, metadata, and indicator catalog.

See `INEClient-fields` for configurable fields (language, caching, timeouts, etc.).

## Data

`get_data(indicator, row_limit, ...)` Retrieve tidy data for an indicator, with automatic chunking and optional caching.

`download_data(indicator, row_limit, ...)` Download data to the file cache without loading into memory.

`load_raw_data(indicator)` Load previously downloaded raw JSON data from the file cache.

`preview_chunks(indicator, row_limit, ...)` Preview how many API chunks a download would require.

## Metadata

`get_metadata(indicator)` Get cleaned metadata for an indicator.

`info(indicator)` Print a summary of an indicator's key properties.

`get_dim_info(indicator)` Get dimension descriptions.

`get_dim_values(indicator, dims)` Get possible values for all dimensions.

`is_valid(indicator)` Check if an indicator exists.

`is_updated(indicator, last_updated, metadata)` Check if an indicator has been updated since last download.

## Catalog

`get_catalog()` Download and parse the full indicator catalog (~10 min).

`download_catalog()` Download the catalog to the file cache.

## Cache

`list_cached()` List indicators present in the file cache.

`clear_cache(indicator)` Clear cached files.

## Active bindings

`lang` Language code (`"PT"` or `"EN"`).

`use_cache` Whether caching is enabled.

`cache_dir` Cache directory path, or `NULL` for default.

`row_limit` Default maximum output rows per API request.

`max_retries` Maximum retry attempts for chunk downloads.

`progress_interval` Print progress every N chunks during downloads.

`timeout` Timeout in seconds for API requests.

## Methods

### Public methods:



**Method** `new()`**:** Create a new INE API client.

*Usage:*

```
INEClient$new(
  lang = "PT",
  use_cache = FALSE,
  cache_dir = NULL,
  row_limit = 1000000L,
  max_retries = 3L,
  progress_interval = 10L,
  timeout = 300
)
```

*Arguments:*

`lang` Language code: `"PT"` (default) or `"EN"`.

`use_cache` Logical. Whether to cache API responses. Default `FALSE`.

`cache_dir` Character or `NULL`. Cache directory path. If `NULL` (default), uses `tools::R_user_dir("ineptr2", "cache")`.

`row_limit` Integer. Default maximum output rows per API request. Default `1000000L`.

`max_retries` Integer. Maximum retry attempts for failed chunk downloads. Default `3L`.

`progress_interval` Integer. Print a progress message every N chunks during downloads. Default `10L`.

`timeout` Numeric. Timeout in seconds for API requests (metadata and data endpoints). Default `300` (5 minutes). The catalog endpoint uses a separate, longer timeout.

*Returns:* A new `INEClient` object.

**Method** `get_data()`**:** Retrieve tidy data for an indicator.

*Usage:*

```
INEClient$get_data(indicator, row_limit = NULL, ...)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

`row_limit` Integer or `NULL`. Maximum output rows per API request before splitting into multiple calls. If `NULL` (default), uses the client's `row_limit` field. See **Details**.

`...` Dimension filters. Each argument should be named `dimN` (where N is the dimension number) with a character vector of values. Omitted dimensions include all values.

*Details:*

*Row limit and chunking:*
The INE API limits each request to **1 000 000 output rows**, counted as the product of unique values across all dimensions. When the estimated row count exceeds `row_limit`, the request is automatically split into smaller chunks by iterating over one or more dimensions.
If requests are timing out, try lowering `row_limit` (or increasing the client's `timeout` field) to produce more, smaller chunks.

*Caching:*
When `use_cache` is enabled, processed data is stored as an RDS file. Subsequent calls with the same or narrower dimension filters return the cached result without hitting the API. Changing filters to include values outside the cached set triggers a fresh download.

*Returns:* A data frame with the indicator data.

**Method** `download_data()`**:** Download data for an indicator to the file cache without loading it into memory. Caching is temporarily enabled for the duration of the call regardless of the client's `use_cache` setting.

*Usage:*

```
INEClient$download_data(indicator, row_limit = NULL, ...)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

`row_limit` Integer or `NULL`. Maximum output rows per API request before splitting into multiple calls. If `NULL` (default), uses the client's `row_limit` field.

`...` Dimension filters in the form `dimN = value`.

*Returns:* Invisibly, a list with `indicator`, `cache_dir`, `total_chunks`, and `complete`, or `invisible(NULL)` on partial download failure (resume by calling again).

**Method** `load_raw_data()`**:** Load previously downloaded raw data from the file cache as a list of parsed JSON responses. Use `download_data()` first to populate the cache.

*Usage:*

```
INEClient$load_raw_data(indicator)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

*Returns:* A list with `responses` (parsed JSON) and `urls`.

**Method** `get_metadata()`**:** Get cleaned metadata for an indicator.

*Usage:*

```
INEClient$get_metadata(indicator)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

*Returns:* API response body as a list.

**Method** `get_catalog()`**:** Get the full INE indicator catalog. This operation is very time-consuming (~10 minutes) as it downloads the entire catalog from the INE API. Consider using `download_catalog()` to cache the result for subsequent calls.

*Usage:*

```
INEClient$get_catalog()
```

*Returns:* A data frame with one row per indicator.

**Method** `download_catalog()`**:** Download the INE indicator catalog to the file cache without loading it into memory. This operation is time-consuming (~10 minutes) as it downloads the entire catalog from the INE API. Subsequent calls return the cached file immediately. Caching is temporarily enabled for the duration of the call regardless of the client's `use_cache` setting.

*Usage:*

```
INEClient$download_catalog()
```

*Returns:* Invisibly, the cache file path.

**Method** `info()`**:** Print a summary of an indicator's key properties: code, name, periodicity and time range, last update date, and a per-dimension breakdown of unique values. Labels are displayed in the client's current language.

*Usage:*

```
INEClient$info(indicator)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

*Returns:* Invisibly, a list with `code`, `name`, `periodicity`, `first_period`, `last_period`, `last_updated`, and `dimensions` (a data frame with `dim_num`, `name`, and `n_values` columns).

**Method** `get_dim_info()`**:** Get dimension descriptions for an indicator.

*Usage:*

```
INEClient$get_dim_info(indicator)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

*Returns:* A data frame with dim_num, abrv, and versao columns.

**Method** `get_dim_values()`**:** Get possible values for all dimensions of an indicator.

*Usage:*

```
INEClient$get_dim_values(indicator, dims = NULL)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

`dims` Integer vector of dimension numbers to include, or `NULL` (default) for all dimensions.

*Returns:* A tidy data frame with dimension values.

**Method** `preview_chunks()`**:** Preview how many API chunks a download would require, without fetching any data. Useful for estimating download time before committing to a large request.

*Usage:*

```
INEClient$preview_chunks(indicator, row_limit = NULL, ...)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

`row_limit` Integer or `NULL`. Maximum output rows per API request before splitting into multiple calls. If `NULL` (default), uses the client's `row_limit` field.

`...` Dimension filters in the form `dimN = value`.

*Returns:* Invisibly, a list with `chunks` and `estimated_rows`.

**Method** `is_valid()`**:** Check if an indicator exists and is callable via the INE API.

*Usage:*

```
INEClient$is_valid(indicator)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

*Returns:* `TRUE` if indicator exists, `FALSE` otherwise.

**Method** `is_updated()`**:** Check if an indicator has been updated since last download.

*Usage:*

```
INEClient$is_updated(indicator, last_updated = NULL, metadata = NULL)
```

*Arguments:*

`indicator` INE indicator ID as a 7-digit string. Example: `"0010003"`.

`last_updated` A `Date` object or a character string in `"YYYY-MM-DD"` format. If provided, takes precedence over cached metadata. If `NULL` (default), the function looks for cached metadata or the `metadata` argument.

`metadata` A metadata list object as returned by `get_metadata()`. If provided and `last_updated` is `NULL`, extracts `DataUltimaAtualizacao`.

*Returns:* TRUE if updated, FALSE if not.

**Method** `list_cached()`**:** List indicators present in the file cache.

*Usage:*

```
INEClient$list_cached()
```

*Returns:* A data frame with one row per cached indicator and columns `indicator`, `has_metadata`, `has_data`, `chunks_downloaded`, `chunks_total`, and `download_complete`. Returns a zero-row data frame if no cache exists.

**Method** `clear_cache()`**:** Clear cached files.

*Usage:*

```
INEClient$clear_cache(indicator = NULL)
```

*Arguments:*

`indicator` Optional INE indicator ID. If NULL (default), clears all cached files.

*Returns:* Invisibly returns TRUE if files were removed, FALSE otherwise.

**Method** `print()`**:** Print a summary of the client configuration.

*Usage:*

```
INEClient$print(...)
```

*Arguments:*

`...` Ignored.

**Method** `clone()`**:** The objects of this class are cloneable with this method.

*Usage:*

```
INEClient$clone(deep = FALSE)
```

*Arguments:*

`deep` Whether to make a deep clone.

## See Also

INEClient-fields for field descriptions.

## Examples

```
# -- Setup --
ine <- INEClient$new()
ine <- INEClient$new(lang = "EN", use_cache = TRUE)
print(ine)

# -- Metadata --
meta <- ine$get_metadata("0010003")
ine$info("0010003")
dims <- ine$get_dim_info("0010003")
vals <- ine$get_dim_values("0010003")

# -- Data --
```

```
df <- ine$get_data("0010003")
df <- ine$get_data("0010003", dim1 = "S7A2024", dim2 = c("11", "17"))
ine$preview_chunks("0008273")

# -- Validation --
ine$is_valid("0010003")
ine$is_updated("0010003", last_updated = "2024-01-01")

# -- Cache --
ine$list_cached()
ine$clear_cache()
```

---

`INEClient-fields` *INEClient configuration fields*

---

## Description

Configuration fields for the INEClient class. All fields are implemented as active bindings with validation. Set them with `ine$field <- value` and read them with `ine$field`.

## Arguments

| | |
|---|---|
| `lang` | Character. Language code: `"PT"` (default) or `"EN"`. Affects API responses, cache file paths, and display labels. |
| `use_cache` | Logical. Whether to cache API responses locally. Default `FALSE`. |
| `cache_dir` | Character or `NULL`. Cache directory path. If `NULL` (default), uses `tools::R_user_dir("ineptr2", "cache")`. |
| `row_limit` | Integer. Maximum output rows per API request before splitting into chunks. Must be between 1 and 1 000 000 (the API ceiling). Default `1000000L`. |
| `max_retries` | Integer. Maximum retry attempts for failed chunk downloads. Default `3L`. |
| `progress_interval` | Integer. Print a progress message every N chunks during downloads. Default `10L`. |
| `timeout` | Numeric. Timeout in seconds for API requests (metadata and data endpoints). Default `300` (5 minutes). The catalog endpoint uses a separate, longer timeout. |

## See Also

INEClient for methods.

## Examples

```
ine <- INEClient$new()
ine$lang
ine$lang <- "EN"

ine$use_cache <- TRUE
ine$cache_dir <- tempdir()
ine$row_limit <- 500000L
```

# Index