Package 'TEMPO'

October 30, 2026

Type Package

Title Tools for Romania's National Institute of Statistics Tempo Data

Version 0.2.0

Description Find, inspect and download statistical tables from the Romanian National Institute of Statistics' TEMPO Online database. Tables can be listed and searched in Romanian or English, their structure inspected before downloading, and whole tables or filtered selections retrieved either as data frames or as CSV files that are kept up to date between sessions.

License GPL-3 | EUPL-1.1

URL https://github.com/RProjectRomania/TEMPO

BugReports https://github.com/RProjectRomania/TEMPO/issues

Depends R (>= 3.6)

Imports curl (>= 5.0.0), jsonlite (>= 1.7.0), stats, tools, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), webfakes (>= 1.1.0), withr

Config/testthat/edition 3

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Config/roxygen2/version 8.1.1

Author Marian Necula [aut, cre], Bogdan Oancea [aut], Ana Maria Tiru [aut]

Maintainer Marian Necula <marian.necula00@gmail.com>

Repository CRAN

Date/Publication 2026-10-30 21:00:02 UTC

Contents

TEMPO-conditions 2
TEMPO-options 3
tempo_bulk 4
tempo_cache_clear 6
tempo_cache_dir 7
tempo_get 8
tempo_hierarchy 9
tempo_meta 11
tempo_search 13
tempo_tidy 14
tempo_toc 16

Index **19**

`TEMPO-conditions` *Errors and warnings raised by TEMPO*

Description

Every error raised by this package carries an S3 class, so a caller can react to the *kind* of failure without matching on English message text. That matters for unattended work: an unreachable service is worth retrying later, whereas a rejected matrix code never will be.

Error classes

All of them inherit from `tempo_error`, and from `error` and `condition`, so an ordinary `tryCatch(expr, error = ...)` still catches everything.

`tempo_input_error` Invalid arguments, detected before any request is made. Carries no extra fields.

`tempo_offline_error` The service could not be reached at all, after exhausting `getOption("TEMPO.max_attempts")`. Field: `url`.

`tempo_http_error` The service answered with a non-2xx status. Fields: `status_code`, `url`, `body` (the first 200 characters of the response).

`tempo_api_error` The service answered, but not with the structure the package expects: malformed JSON, a missing field, inconsistent CSV headers across chunked requests. Field: `code`, when a matrix is involved.

`tempo_io_error` A local path could not be created, written or read. Field: `path`.

Warning classes

Warnings inherit from `tempo_warning` and are raised at most once per session each, so a loop over many matrices does not bury the console.

`tempo_date_warning` The service reported an update date in a format this package cannot parse. `Last_update` becomes `NA`; freshness checks fall back to comparing the service's raw update string, so caching keeps working.

`tempo_cache_warning` A CSV file is being reused although its language cannot be verified, because it predates the download manifest.

`tempo_tidy_warning` `tempo_tidy()` met something it could not interpret, such as an observation that is not a number.

Examples

```
# Distinguish a bad argument from a service problem.
result <- tryCatch(
  tempo_toc(language = "fr"),
  tempo_input_error = function(condition) "fix the call",
  tempo_offline_error = function(condition) "retry later",
  tempo_error = function(condition) "something else"
)
result

## Not run:
# Retry only what is worth retrying, and read the status code.
tryCatch(
  tempo_get("ACC101B"),
  tempo_http_error = function(condition) {
    message("HTTP ", condition$status_code, " from ", condition$url)
    NULL
  },
  tempo_offline_error = function(condition) {
    Sys.sleep(60)
    tempo_get("ACC101B")
  }
)

## End(Not run)
```

`TEMPO-options` *Options that control TEMPO*

Description

`TEMPO` is configured with `base::options()`. Every option is named with a `TEMPO.` prefix and falls back to the default shown below when unset.

Connection

`TEMPO.base_url` Base URL of the TEMPO Online service. Default `"http://statistici.insse.ro:8077/tempo-ins"`. The service is only reachable over plain HTTP; it does not offer TLS.

`TEMPO.connecttimeout` Seconds to wait for the connection. Default `30`.

`TEMPO.timeout` Seconds to wait for a complete response. Default `300`.

`TEMPO.user_agent` `User-Agent` header sent with every request. Default `NULL`, which sends a string identifying the package, its version and the R version.

Politeness and resilience

`TEMPO.max_attempts` How many times a failed request is attempted in total. Default `3`. Transport failures and HTTP 408, 425, 429 and 5xx statuses are retried; other statuses fail immediately.

`TEMPO.backoff` Base number of seconds for the exponential backoff between attempts. Default `1`, which waits roughly 1 then 2 seconds. A `Retry-After` response header takes precedence.

`TEMPO.throttle` Minimum number of seconds between two consecutive requests. Default `0`.

`TEMPO.max_connections` Maximum number of concurrent connections used when many requests are issued at once, as in `tempo_toc()` with `full_description = TRUE`. Default `4`. TEMPO Online is a small public service; please do not raise this casually.

Query size

`TEMPO.max_cells` Largest number of cells requested in a single query. Default `50000`. Larger downloads are split into several requests.

`TEMPO.max_options_per_dim` Largest number of options selected from a single dimension in one query. Default `300`.

`TEMPO.max_requests` Refuse to start a download that would need more than this many requests. Default `2000`.

Output

`TEMPO.progress` Whether to report progress for long operations. Default `NULL`, meaning `base::interactive()`.

`TEMPO.cache_dir` Directory used to cache service responses. See `tempo_cache_dir()`.

Examples

```
# Be a little gentler with the service and allow bigger single queries.
old <- options(TEMPO.throttle = 0.2, TEMPO.max_cells = 1e5)
getOption("TEMPO.throttle")
options(old)
```

---

tempo_bulk *Download complete TEMPO Online tables*

Description

Downloads one or more complete matrices from the Romanian National Institute of Statistics TEMPO Online database and writes each as a CSV file. Use `tempo_get()` instead when you want part of a table, or the data in memory.

Usage

```
tempo_bulk(
  codes,
  language = "ro",
  directory,
  overwrite = FALSE,
  progress = NULL
)
```

Arguments

codes	A character vector (or list) of TEMPO matrix codes, such as "ACC101B". Use tempo_toc() or tempo_search() to discover codes.
language	A single string: "ro" for Romanian (the default) or "en" for English.
directory	A single path for the downloaded CSV files. It is created when needed, and nothing outside it is ever written.
overwrite	A single logical value. If TRUE, download every requested matrix even when the local copy is already current. Default FALSE.
progress	A single logical value, or NULL (the default) to follow getOption("TEMPO.progress"), which in turn defaults to base::interactive().

Value

Invisibly, a character vector of the CSV paths of **all** requested matrices, named by matrix code. Three attributes describe what happened:

downloaded Logical, whether the file was fetched in this call.

reason Why each file was or was not fetched: "missing", "outdated", "revised", "language", "modified", "unverifiable", "overwrite" or "current".

requests Number of HTTP requests spent on each matrix.

Up to and including version 0.1.0 only the newly written paths were returned, which forced callers to rebuild the paths of skipped files by hand.

Freshness, and why a manifest

Alongside the CSV files, tempo_bulk() maintains .tempo-manifest.json, recording for every file the language, the service's own update string, the retrieval time, the number of requests and an MD5 checksum. A file is reused only when all of the language, the published revision and the checksum agree.

A CSV file does not record which language it holds, so the earlier modification-time test accepted a Romanian file for an English request and reported it as up to date. Files written before the manifest existed are still accepted on the old modification-time rule, with a warning that their language cannot be verified; delete them, or pass overwrite = TRUE, to be certain.

Query size

A full matrix can exceed what the service will answer in one query, so the request is split into chunks whose product of selected values stays within `getOption("TEMPO.max_cells")`. The parts are concatenated, and each part's CSV header is checked against the first part's so that misaligned columns are an error rather than a corrupt file. See TEMPO-options.

See Also

`tempo_get()` for filtered, in-memory downloads, `tempo_tidy()` to turn a downloaded CSV into analysis-ready data, `tempo_meta()` to size a table first.

Examples

```
## Not run:
data_directory <- file.path(tempdir(), "tempo-data")

files <- tempo_bulk("ACC101B", language = "en", directory = data_directory)
files
attr(files, "reason")

# A second call reuses the file and says so.
tempo_bulk("ACC101B", language = "en", directory = data_directory)

# Paths are returned for skipped files too, so this always works.
accidents <- tempo_tidy(files[["ACC101B"]])
head(accidents)

## End(Not run)
```

---

tempo_cache_clear *Delete cached TEMPO responses*

Description

Removes every cached response from `tempo_cache_dir()`. Use it after the service has published new data and you want to be certain that nothing stale is reused.

Usage

```
tempo_cache_clear()
```

Value

The number of cache entries removed, invisibly.

See Also

`tempo_cache_dir()`

Examples

```
tempo_cache_clear()
```

`tempo_cache_dir` *Where TEMPO caches service responses*

Description

The table of contents and the per-matrix metadata change rarely but are expensive to fetch, so `TEMPO` caches them. Call `tempo_cache_dir()` to see the directory in use and `tempo_cache_dir(path)` to change it.

Usage

```
tempo_cache_dir(path = NULL)
```

Arguments

`path`	A single directory path, or `NULL` to query the current setting without changing it. The directory is created when it does not exist.

Details

By default the cache lives in a subdirectory of `base::tempdir()`, so it is discarded when the R session ends and nothing is written outside the session temporary area. To keep the cache between sessions, point it at a persistent location; `tools::R_user_dir()` gives the conventional one:

```
tempo_cache_dir(tools::R_user_dir("TEMPO", "cache"))
```

Equivalently, set `options(TEMPO.cache_dir = ...)`, which is useful in a project `.Rprofile`. Entries older than `TEMPO.cache_max_age` seconds (one day by default) are ignored and refetched.

Value

The cache directory, invisibly when setting it.

See Also

`tempo_cache_clear()`, TEMPO-options

Examples

```
# The default cache is inside the session temporary directory.
tempo_cache_dir()

# Use a project-local cache instead.
old <- tempo_cache_dir(file.path(tempdir(), "my-tempo-cache"))
tempo_cache_dir()
tempo_cache_dir(old)
```

tempo_get *Download part of a TEMPO Online table into R*

Description

Fetches a matrix, or a selection from it, and returns it as a data frame. Nothing is written to disk. Unlike `tempo_bulk()`, which always downloads a whole table to CSV, `tempo_get()` sends the dimension selection to the service, so asking for one county and ten years costs one small request rather than a download of the entire table.

Usage

```
tempo_get(
  code,
  filters = NULL,
  language = "ro",
  tidy = TRUE,
  cache = TRUE,
  progress = NULL,
  ...
)
```

Arguments

`code`	A single TEMPO matrix code, such as `"ACC101B"`.
`filters`	A named list selecting values from one or more dimensions, or `NULL` (the default) for the complete table. Names are dimension names, as reported by `tempo_meta()`; a partial, case-insensitive name or a dimension position also works. Dimensions you do not mention, or set to `NULL`, are taken whole. See Matching below.
`language`	A single string: `"ro"` for Romanian (the default) or `"en"` for English. Dimension names, value labels and the values you filter on are all in this language.
`tidy`	A single logical value. If `TRUE` (the default), the result is passed through `tempo_tidy()`: a numeric `value` column, the unit of measure as a column, parsed periods and no footer rows. Set `FALSE` to get the service's own CSV layout, every column a character vector.
`cache`	A single logical value, passed to `tempo_meta()` for the metadata lookup. The data itself is never cached. Default `TRUE`.
`progress`	A single logical value, or `NULL` (the default) to follow `getOption("TEMPO.progress")`.
`...`	Further arguments passed to `tempo_tidy()`, such as `periods = FALSE` or `clean_names = TRUE`.

Value

A data frame. These attributes record the provenance of the result: `tempo_code`, `tempo_language`, `tempo_requests`, `tempo_service_update` and `tempo_retrieved_at`.

Matching

Each element of a filter is matched against the dimension's value labels, in three steps: an exact match ignoring case and surrounding space; then a whole-token match, so `2020` finds `"Anul 2020"`; then an exact match against the service's own numeric value identifiers. An element that matches nothing is an error listing the available values, rather than a silently smaller result.

Size

A selection still has to fit the service's query limits, and a large one is split into several requests exactly as in `tempo_bulk()`. Call `tempo_meta()` first to see how many cells and requests a selection implies.

See Also

`tempo_meta()` for the dimensions available to filter on, `tempo_bulk()` for whole tables as CSV files, `tempo_tidy()` for the cleaning rules.

Examples

```
## Not run:
# What can be filtered?
tempo_meta("ACC101B", language = "en")

# The whole table, tidied, in memory.
accidents <- tempo_get("ACC101B", language = "en")
head(accidents)

# One county and a decade: a small request instead of a full download.
cluj <- tempo_get(
  "ACC101B",
  filters = list(Counties = "Cluj", Years = 2015:2024),
  language = "en"
)
head(cluj)

# The service's own layout, untouched.
raw <- tempo_get("ACC101B", language = "en", tidy = FALSE)
str(raw)

## End(Not run)
```

---

tempo_hierarchy *Retrieve the TEMPO Online classification tree*

Description

Returns the service's own navigation tree: the statistical domains, sub-domains and surveys, and optionally every matrix beneath them, each with its parent and its full classification path.

Usage

```
tempo_hierarchy(language = "ro", matrices = TRUE, cache = TRUE)
```

Arguments

language	A single string: "ro" for Romanian (the default) or "en" for English.
matrices	A single logical value. If TRUE (the default), also list the matrices under each survey. The service does not include them in the tree listing, so this costs one additional request per deepest branch, currently 262 of them; they are issued concurrently and cached. Set FALSE for the branch structure alone, which is one request.
cache	A single logical value. If TRUE (the default), reuse and populate the response cache described in tempo_cache_dir().

Details

This is the cheap way to learn where a matrix sits in the classification. The branch structure arrives in a single request. tempo_toc(full_description = TRUE) recovers the same three levels by requesting the metadata of roughly 1500 matrices, one at a time, because it also reports each matrix's publication date.

Value

A data frame with one row per node and the columns

code Node code. For a matrix this is the code accepted by tempo_get() and tempo_bulk().

name Node label, as plain text.

parent_code Code of the parent node, NA at the root.

level Depth, 1 for a statistical domain.

is_matrix TRUE for statistical matrices.

path Full classification path, levels separated by " > ".

Stability

TEMPO Online publishes no API contract. The two response shapes this function reads have been confirmed against the live service, and both the field names and the wrapper are matched tolerantly, but a change at the service could still break the walk. It then raises a tempo_api_error rather than returning something half-built, and tempo_toc(full_description = TRUE) remains available as the longer-established route to the same three classification levels.

See Also

tempo_toc(), tempo_search(), tempo_meta()

Examples

```
## Not run:
# One request: the classification itself.
branches <- tempo_hierarchy(language = "en", matrices = FALSE)
table(branches$level)
subset(branches, level == 1, select = c(code, name))

# The whole tree, matrices included.
tree <- tempo_hierarchy(language = "en")

# Where does one matrix sit?
tree[tree$code == "POP105A", c("level", "path")]

# Every matrix under one domain, without describing all 1500 tables.
population <- tree[tree$is_matrix & grepl("POPULATIE", tree$path), ]
head(population[, c("code", "name")])

## End(Not run)
```

`tempo_meta` *Describe a TEMPO matrix without downloading it*

Description

Returns the structure of a single matrix: its dimensions, the selectable values of each dimension, the publication date, the position in the statistical classification, and how large a full download would be. This is the information needed to choose sensible `filters` for `tempo_get()`, and it answers "how big is this table?" in one request instead of a download.

Usage

```
tempo_meta(code, language = "ro", cache = TRUE)

## S3 method for class 'tempo_meta'
print(x, ...)
```

Arguments

`code`	A single TEMPO matrix code, such as `"ACC101B"`. Use `tempo_toc()` or `tempo_search()` to find one.
`language`	A single string: `"ro"` for Romanian (the default) or `"en"` for English. Dimension and value labels are returned in this language.
`cache`	A single logical value. If `TRUE` (the default), reuse and populate the response cache described in `tempo_cache_dir()`.
`x`	A `tempo_meta` object.
`...`	Ignored.

Value

An object of class "tempo_meta": a list with components

code, name, language Identification.

last_update Publication date as a Date, or NA when the service reports a date this package cannot parse.

service_update The service's own update string, verbatim.

domain, sub_domain, survey Classification, as plain text.

dimensions A data frame with one row per dimension, giving its position, name and number of values.

values A named list with one data frame per dimension, each holding the id and label of every selectable value.

cells Number of cells a full download would request.

requests Number of HTTP requests a full download would need under the current query-size options.

details The service's query parameters (matMaxDim, matUMSpec).

The unparsed service response is attached as the "raw" attribute.

The unit of measure

The service exposes the unit of measure as a dimension of its own, which is why a matrix measured in persons reports a final dimension such as MU: Persons with a single value. It is part of the query, so it appears here; tempo_tidy() moves it out of the data into a unit column.

Undocumented field names

TEMPO Online has no published API contract. The dimension and value labels are read from the first of several plausible field names (label, name, nomItemName, text), and dimensions fall back to dimension_1, dimension_2, ... if none of them is present. The value identifiers, which the query payload actually needs, come from the documented-by-use nomItemId field and are validated.

See Also

tempo_get() to download a subset, tempo_bulk() for whole tables, tempo_toc() to list matrices.

Examples

```
## Not run:
meta <- tempo_meta("ACC101B", language = "en")
meta

# What can be filtered, and how large is the table?
meta$dimensions
meta$cells

# The selectable values of the first dimension.
head(meta$values[[1]])
```

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

`tempo_search` *Search the TEMPO table of contents*

Description

A convenience wrapper around `tempo_toc()` for the common task of finding the matrix codes whose name matches a pattern.

Usage

```
tempo_search(
  pattern,
  language = "ro",
  fields = NULL,
  ignore_case = TRUE,
  fixed = FALSE,
  toc = NULL
)
```

Arguments

`pattern`	A regular expression, or a fixed string when `fixed = TRUE`. Several patterns are combined with a logical OR.
`language`	A single string: `"ro"` for Romanian (the default) or `"en"` for English. The service stores table names without diacritics, so search for the unaccented spelling, as in `"sanatate"`.
`fields`	Character vector of columns to search. Defaults to all character columns of `toc`.
`ignore_case`	A single logical value, `TRUE` by default.
`fixed`	A single logical value. If `TRUE`, match `pattern` literally rather than as a regular expression. Default `FALSE`.
`toc`	An existing table of contents, as returned by `tempo_toc()`. Pass one to search a table you already hold, including a fully described one, instead of fetching it again.

Value

The matching rows of `toc`, with row names reset.

See Also

`tempo_toc()`

Examples

```
## Not run:
# Everything about population, in English.
tempo_search("population", language = "en")

# Two topics at once, searching a table of contents fetched once.
tables <- tempo_toc(language = "en")
tempo_search(c("unemployment", "employment"), toc = tables)

# Search the described table of contents by statistical domain.
described <- tempo_toc(full_description = TRUE, language = "en")
tempo_search("AGRICULTURE", fields = "Statistical_domain", toc = described)

## End(Not run)
```

`tempo_tidy` *Turn a downloaded TEMPO table into analysis-ready data*

Description

The CSV that TEMPO Online returns is not directly usable: the observation column is text, the unit of measure hides in a column *header*, years read `"Anul 2020"` rather than `2020`, and missing observations are marked `":"`. `tempo_tidy()` resolves all of that and leaves one row per observation.

Usage

```
tempo_tidy(
  x,
  code = NULL,
  periods = TRUE,
  na = c(":", ""),
  drop_notes = TRUE,
  clean_names = FALSE,
  value_name = "value",
  unit_name = "unit"
)
```

Arguments

`x`	A data frame as returned by `tempo_get()` with `tidy = FALSE`, the path of a CSV file written by `tempo_bulk()`, or a single string holding CSV text.
`code`	Optional matrix code, recorded on the result as the `tempo_code` attribute.
`periods`	A single logical value. If `TRUE` (the default), parse the time dimensions. See Time layouts below. Default `TRUE`.
`na`	Character vector of strings that mark a missing observation. Defaults to `c(":", "")`, the service's own markers.

`drop_notes` A single logical value. If `TRUE` (the default), drop rows in which every dimension column is empty. Those are the footnote and copyright lines the service appends after the data; rows with a genuine missing observation are kept, because their dimension columns are filled.

`clean_names` A single logical value. If `TRUE`, pass the column names through `base::make.names()` so they can be used unquoted. Default `FALSE`, which keeps the service's own names.

`value_name`, `unit_name` Names for the observation and unit-of-measure columns. Default `"value"` and `"unit"`.

Value

A data frame with one row per observation: the dimension columns as received, `unit` holding the unit of measure, `value` as a numeric vector, and `period` when a sub-annual period could be reconstructed. The attributes `tempo_code` and `tempo_unit` are set when known.

Time layouts

TEMPO uses two, and they are handled differently.

Some matrices carry a dedicated `Ani/Years` dimension, optionally alongside `Luni/Months`, `Trimestre` or `Semestre`. The year column becomes an integer, the sub-annual labels are left readable, and a `period` column of class `Date` is added when the two can be combined.

Others carry a single `Perioade/Periods` dimension whose labels mix frequencies, as in `"Year 2010"`, `"January 2010"`, `"February 2010"`. Here the label is left exactly as received, because it is the only thing distinguishing an annual total from a monthly figure, and two columns are added instead: `year`, and `period` holding the first day of the month, quarter or semester named. `period` is `NA` for an annual row, so `subset(x, !is.na(period))` selects the sub-annual rows and `subset(x, is.na(period))` the annual ones.

What changed since 0.1.0

Version 0.1.0 shipped no cleaning at all; every caller rewrote these rules. Earlier versions had a `tempo_clean()` which did part of the work but assigned its result into the global environment as a side effect, so it could not be used inside a pipeline. `tempo_tidy()` is a plain function of its input: it returns a value and modifies nothing.

Unparsed observations

Values that are neither numeric nor listed in `na` are reported once with a warning, and the original text is kept in a `value_raw` column so nothing is lost. This is how an unexpected flag or thousands separator becomes visible instead of turning quietly into `NA`.

See Also

`tempo_get()`, which applies this by default, and `tempo_bulk()`.

Examples

```
# The structure the service returns, including a footer line and a
# missing observation marked ":".
csv <- paste(
  "Judete,Ani,UM: Numar persoane,Valoare",
  "Cluj,Anul 2019,Numar persoane,1234",
  "Cluj,Anul 2020,Numar persoane,:",
  "Bihor,Anul 2019,Numar persoane,987",
  ",,,(c) INS 1998-2026",
  sep = "\n"
)

tidy <- tempo_tidy(csv, code = "DEMO101A")
tidy
str(tidy)

# A combined Periods dimension: the annual total stays distinguishable
# from the monthly figures, which share its year.
mixed <- paste(
  "Counties,Periods,UM: Persons,Value",
  "Cluj,Year 2010,Number of persons,100",
  "Cluj,January 2010,Number of persons,8",
  "Cluj,February 2010,Number of persons,9",
  sep = "\n"
)
tempo_tidy(mixed)
subset(tempo_tidy(mixed), is.na(period))   # the annual row

# Monthly labels gain a Date column.
monthly <- paste(
  "Luni,Ani,UM: Numar,Valoare",
  "luna ianuarie,Anul 2024,Numar,10",
  "luna februarie,Anul 2024,Numar,12",
  sep = "\n"
)
tempo_tidy(monthly)

## Not run:
# From a file written by tempo_bulk().
files <- tempo_bulk("ACC101B", language = "en",
                    directory = file.path(tempdir(), "tempo-data"))
head(tempo_tidy(files[["ACC101B"]], code = "ACC101B"))

## End(Not run)
```

---

tempo_toc *List tables available from TEMPO Online*

Description

Retrieves the table of contents for the Romanian National Institute of Statistics TEMPO Online database: one row per statistical matrix, with the matrix code needed by `tempo_bulk()`, `tempo_get()` and `tempo_meta()`.

Usage

```
tempo_toc(full_description = FALSE, language = "ro", cache = TRUE)
```

Arguments

`full_description`	A single logical value. If `TRUE`, also retrieve the statistical domain, sub-domain, survey name and last-update date of every table. This needs one request per table, which the service answers for roughly 1500 matrices, so the first call takes a while even though the requests are issued concurrently and cached afterwards. Default `FALSE`.
`language`	A single string: `"ro"` for Romanian (the default) or `"en"` for English.
`cache`	A single logical value. If `TRUE` (the default), reuse and populate the response cache described in `tempo_cache_dir()`.

Value

A data frame with one row per matrix and the columns

`name` Table name in the requested language.

`code` Matrix code, for example `"ACC101B"`.

`comment` Service comment, when the service supplies any.

With `full_description = TRUE` it additionally contains

`Statistical_domain`, `Statistical_sub_domain`, `Survey_name` The first three levels of the service's classification, as plain text.

`Press_releases` Press-release URLs that the service embeds in the domain label, space separated, or `NA`.

`Last_update` Publication date of the matrix, class `Date`.

Text and HTML

The service embeds HTML anchors in some labels, most visibly in the top-level statistical domains, which arrive as `AGRICULTURE <a href="https://insse.ro/...">Press releases</a>`. Those tags are stripped from every character column, and the URLs are kept separately in `Press_releases`. Table names are returned without Romanian diacritics, because that is how the service stores them.

Errors

The function needs an internet connection and raises classed conditions, so a caller can distinguish an unreachable service from a malformed response. See TEMPO-conditions.

See Also

`tempo_search()` to filter the result, `tempo_meta()` for the full metadata of a single matrix, `tempo_hierarchy()` for the classification tree.

Examples

```
## Not run:
tables <- tempo_toc(language = "en")
head(tables)
nrow(tables)

# The slow, fully described variant. Cached, so a second call is cheap.
described <- tempo_toc(full_description = TRUE, language = "en")
head(described[order(described$Last_update, decreasing = TRUE), ])

## End(Not run)
```

Index

base::interactive(), *4, 5*
base::make.names(), *15*
base::tempdir(), *7*

print.tempo_meta *(*tempo_meta*)*, 11

TEMPO-conditions, 2, *17*
TEMPO-options, 3, *6, 7*
tempo_bulk, 4
tempo_bulk(), *8–10*, *12*, *14, 15*, *17*
tempo_cache_clear, 6
tempo_cache_clear(), *7*
tempo_cache_dir, 7
tempo_cache_dir(), *4*, *6*, *10, 11*, *17*
tempo_get, 8
tempo_get(), *4*, *6*, *10–12*, *14, 15*, *17*
tempo_hierarchy, 9
tempo_hierarchy(), *18*
tempo_meta, 11
tempo_meta(), *6*, *8–10*, *17, 18*
tempo_search, 13
tempo_search(), *5*, *10, 11*, *18*
tempo_tidy, 14
tempo_tidy(), *3*, *6*, *8, 9*, *12*
tempo_toc, 16
tempo_toc(), *4, 5*, *10–13*
tools::R_user_dir(), *7*