Package: datagouvfr (via r-universe)

May 25, 2026

Type Package

Title Access and Query French Government Open Data from data.gouv.fr

Version 0.1.0

Description Provides functions to search, retrieve metadata, and download datasets from https://data.gouv.fr, the official French government open data portal. Includes tools for querying datasets with filtering capabilities, automatic caching of downloaded resources, and flexible access methods using both direct CSV downloads and the data.gouv.fr tabular API.

License file LICENSE

URL https://inrae.r-universe.dev/datagouvfr

BugReports https://forge.inrae.fr/umr-g-eau/datagouvfr/-/issues

Encoding UTF-8

LazyData true

Imports data.table, dplyr, httr, logger, lubridate, purrr, R.utils, rjson, rvest, stats, stringr, tibble, urltools, utils

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev libxml2-dev libssl-dev

Repository https://inrae.r-universe.dev

Date/Publication 2026-05-25 13:23:39 UTC

RemoteUrl https://forge.inrae.fr/umr-g-eau/datagouvfr

RemoteRef HEAD

RemoteSha d63de91390f63b22c7c26f78eb34fc70277bebac

Contents

`convert_list_to_tibble` *Convert list provided by the APIs into a tibble*

Description

Convert list provided by the APIs into a tibble

Usage

```
convert_list_to_tibble(l)
```

Arguments

l	a [list] provided by the API (See [query_api])

Details

This function is used internally by all the retrieving data functions for converting data after the call to [query_api].

Value

A [tibble::tibble] with one row by record and one column by field.

Examples

```
# Get last meteo data around Espelette from the API (Lambert coords are in hm)
df <- query_api(resource_id = get_latest_sim2_resource_id(),
                LAMBX__greater = 2750,
                LAMBX__less = 3040,
                LAMBY__greater = 18100,
                LAMBY__less = 18400)
df

# Get last meteo data around Espelette from the CSV file (Lambert coords are in hm)
df <- query_csv(resource_id = get_latest_sim2_resource_id(),
                LAMBX__greater = 2750,
                LAMBX__less = 3040,
                LAMBY__greater = 18100,
                LAMBY__less = 18400)
df

# Get last meteo data around Espelette from the API if available (Lambert coords are in hm)
```

```
df <- query(resource_metadata = get_latest_sim2_resource_id(metadata = TRUE),
            LAMBX__greater = 2750,
            LAMBX__less = 3040,
            LAMBY__greater = 18100,
            LAMBY__less = 18400)
df
```

---

download_resource *Query resource data from data.gouv.fr*

Description

Download and filter data depending on their available format:

Usage

```
download_resource(
  resource_id,
  resource_metadata = get_resource_metadata(resource_id),
  url_pattern = "https://www.data.gouv.fr/fr/datasets/r/%s",
  cache_dir = Sys.getenv("DATAGOUVFR_CACHE_DIR", file.path(dirname(tempdir()),
    "datagouvfr")),
  force_download = FALSE
)

query(resource_metadata, ..., force_download = FALSE)

query_api(
  resource_id,
  ...,
  url_pattern = "https://tabular-api.data.gouv.fr/api/resources/%s/data/",
  raw_format = FALSE
)

query_csv(
  resource_id,
  resource_metadata = get_resource_metadata(resource_id),
  ...,
  url_pattern = "https://www.data.gouv.fr/fr/datasets/r/%s",
  cache_dir = Sys.getenv("DATAGOUVFR_CACHE_DIR", file.path(dirname(tempdir()),
    "datagouvfr")),
  force_download = FALSE
)
```

Arguments

resource_id	resource ID (See [get_resources_id()])

resource_metadata	resource metadata (one item of the list returned by the function [get_resources_metadata])
url_pattern	URL pattern to get data from the API (injected in [sprintf] with the resource ID to complete the URL)
cache_dir	folder where resources are downloaded. It uses the value stored in the environment variable 'DATAGOUVFR_CACHE_DIR', or the system temporary folder if the later is not defined
force_download	force download instead of using cache for 'query_csv'
...	filter parameters (See details)
raw_format	if 'TRUE' the API response is not formatted as [tibble]

Details

- 'query_csv': download and cache a tabular file in CSV format and filter it - 'query_api': directly query the [data.gouv.fr tabular API](https://www.data.gouv.fr/en/dataservices/api-tabulaire-data-gouv-fr-beta/) - 'query': automatically launch 'query_csv' or 'query_api' depending on the availability of the tabular API given by the resource metadata

'...' are filter parameters that depend on the resource retrieved. Available filter are (replace 'column_name' by the name of the column):

- exact value: 'column_name__exact=value'

'url_pattern' is the URL of the api requested by the data.gouv.fr for displaying the resources. It is injected in [sprintf] with the resource ID to complete the URL.

Value

A [tibble] containing the requested data or a [list] if 'query_api' has its argument 'raw_format' sets to 'TRUE'.

Examples

```
# Get last meteo data around Espelette from the API (Lambert coords are in hm)
df <- query_api(resource_id = get_latest_sim2_resource_id(),
                LAMBX__greater = 2750,
                LAMBX__less = 3040,
                LAMBY__greater = 18100,
                LAMBY__less = 18400)
df

# Get last meteo data around Espelette from the CSV file (Lambert coords are in hm)
df <- query_csv(resource_id = get_latest_sim2_resource_id(),
                LAMBX__greater = 2750,
                LAMBX__less = 3040,
                LAMBY__greater = 18100,
                LAMBY__less = 18400)
df

# Get last meteo data around Espelette from the API if available (Lambert coords are in hm)
df <- query(resource_metadata = get_latest_sim2_resource_id(metadata = TRUE),
```

```
              LAMBX__greater = 2750,
              LAMBX__less = 3040,
              LAMBY__greater = 18100,
              LAMBY__less = 18400)
df
```

`get_dataset_id` *Get resource ID from dataset*

Description

This function fetches the dataset id from the web page base_url/dataset.

Usage

```
get_dataset_id(
  dataset,
  base_url = "https://www.data.gouv.fr/fr/datasets",
  url = file.path(base_url, dataset, "informations")
)
```

Arguments

`dataset`	path of the dataset
`base_url`	URL of the data.gouv.fr datasets repository
`url`	complete url of the dataset (by default base_url/dataset)

Value

The dataset ID

Examples

```
# Get the ID of the SIM2 dataset
get_dataset_id("donnees-changement-climatique-sim-quotidienne")
```

get_latest_sim2_resource_id

Get the latest resource id of a dataset

Description

Get the latest resource id of a dataset

Usage

```
get_latest_sim2_resource_id(
  resources_metadata = get_resources_metadata(dataset_id),
  dataset_id = "6569b27598256cc583c917a7",
  metadata = FALSE
)
```

Arguments

resources_metadata	resource metadata where to fetch latest resource available
dataset_id	dataset ID (See [get_dataset_id()], SIM2 dataset ID is used by default)
metadata	[logical] returns the complete resource metadata instead of only the resource id.

Value

The latest resource ID or metadata [list] depending on ‘metadata‘ argument.

Examples

```
get_latest_sim2_resource_id()
```

get_resources_metadata

Get dataset or resources metadata from dataset ID

Description

Get dataset or resources metadata from dataset ID

Usage

```
get_resources_metadata(
  dataset_id,
 api_pattern = file.path("https://www.data.gouv.fr/api/2/datasets/%s/resources",
    "?page=1&type=main&page_size=6&q=")
)

get_resource_metadata(
  resource_id,
  api_pattern = "https://www.data.gouv.fr/api/2/datasets/resources/%s/"
)

get_dataset_metadata(
  dataset_id,
  api_pattern = "https://www.data.gouv.fr/api/2/datasets/%s/"
)
```

Arguments

dataset_id	Dataset ID (See [get_dataset_id()])
api_pattern	API pattern to get resources metadata (See details)
resource_id	Resource ID

Details

'api_pattern' is the URL of the api requested by the data.gouv.fr for displaying the resources. It is injected in [sprintf] with the dataset ID to complete the URL.

Value

A list of metadata

Examples

```
# Get metadata from SIM2 daily dataset
dataset_id <- get_dataset_id("donnees-changement-climatique-sim-quotidienne")
dataset_id

dataset_metadata <- get_dataset_metadata(dataset_id)
str(dataset_metadata)

resources_metadata <- get_resources_metadata(dataset_id)
str(resources_metadata)
```

get_sim2_data *Get SIM2 data from a period and a rectangular window*

Description

Get SIM2 data from a period and a rectangular window

Usage

```
get_sim2_data(
  date_start = as.Date("1958-08-01"),
  date_end = Sys.Date(),
  ...,
 sim2_selected_meta = get_sim2_resources_metadata_from_date(date_start = date_start,
    date_end = date_end, sim2_metadata =
    get_resources_metadata("6569b27598256cc583c917a7")),
  cache_dir = Sys.getenv("DATAGOUVFR_CACHE_DIR", file.path(dirname(tempdir()),
    "datagouvfr"))
)
```

Arguments

date_start	Start date of the period
date_end	End date of the period
...	Parameters passed to [query]
sim2_selected_meta	A tibble with the metadata of the SIM2 resources to download. (See [get_sim2_resources_metadata_from_
cache_dir	folder where resources are downloaded. It uses the value stored in the environment variable 'DATAGOUVFR_CACHE_DIR', or the system temporary folder if the later is not defined

Details

Be careful, due to the structure of the data, the CSV files downloaded contains data for the whole France territory. 10 years of data correspond to about 1.1 GB to download. However these CSV files are only downloaded once and stored to the folder defined by the parameter 'cache_dir'.

Value

A [tibble] with one row by time step and by cell.

Examples

```
# Get meteorological data of the last 3 months on Espelette territory
data <- get_sim2_data(
  date_start = lubridate::`%m-%`(Sys.Date(), months(3)),
  LAMBX__greater = 2750,
```

```
  LAMBX__less = 3040,
  LAMBY__greater = 18100,
  LAMBY__less = 18400
)
summary(data)
```

`get_sim2_resources_metadata_from_date` *Get resources ID from a period*

Description

This function is particularly adapted for the SIM2 dataset which has resources classified by periods.

The function [get_sim2_resources_periods] returns the periods corresponding to a list of SIM2 resources.

Usage

```
get_sim2_resources_metadata_from_date(
  date_start = as.Date("1958-08-01"),
  date_end = Sys.Date(),
  sim2_metadata = get_resources_metadata("6569b27598256cc583c917a7")
)

get_sim2_resources_periods(
  sim2_metadata = get_resources_metadata("6569b27598256cc583c917a7")
)
```

Arguments

`date_start`	Start date of the period
`date_end`	End date of the period
`sim2_metadata`	Metadata of the SIM2 dataset (See [get_resources_metadata()], SIM2 dataset is used by default)

Value

The selected resources IDs. It also contains an attribute ‘"periods"‘ which contains the start and end dates of each resource.

Examples

```
# What periods are covered by each SIM2 resource?
str(get_sim2_resources_periods())

# Select resources for data since 1990
metadata <- get_sim2_resources_metadata_from_date(date_start = as.Date("1990-01-01"))
names(metadata)
str(lapply(metadata, attr, which = "period"))
```

Index