Package 'meteospain'

January 8, 2026

Title Access to Spanish Meteorological Stations Services

Version 0.3.1

Description
Access to different Spanish meteorological stations data services and APIs (AEMET, SMC, MG, Meteoclimatic...).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.3.0)

Suggests ggforce, ggplot2, keyring, knitr, rmarkdown, testthat (>= 3.0.0), withr

Imports assertthat, dplyr (>= 1.0.0), glue, jsonlite, lubridate, purrr (>= 1.0.0), rlang, sf, stringr, tidyr, units, xml2, vctrs, cachem, cli, httr2, rvest

Config/testthat/edition 3

Config/testthat/parallel true

VignetteBuilder knitr

URL https://emf-creaf.github.io/meteospain/, https://github.com/emf-creaf/meteospain

BugReports https://github.com/emf-creaf/meteospain/issues

NeedsCompilation no

Author Victor Granda [aut, cre] (ORCID: <https://orcid.org/0000-0002-0469-1991>), Miquel de Caceres [ctb] (ORCID: <https://orcid.org/0000-0001-7132-2080>), Aitor Ameztegui [ctb] (ORCID: <https://orcid.org/0000-0003-2006-1559>), Luis Franco [ctb], Rubén Fernández-Casal [ctb]

Maintainer Victor Granda <victorgrandagarcia@gmail.com>

Repository CRAN

Date/Publication 2026-01-08 15:10:08 UTC

Contents

`clear_meteospain_cache` *Clear all cached results*

Description

Reset the internal cache used to limit the API requests.

Usage

```
clear_meteospain_cache()
```

Details

Cached results reduces the number of API requests, but sometimes we need fresh results without restarting the R session. `clear_meteospain_cache` function reset the cache for the actual R session.

`get_meteo_from` *Get meteorological stations data*

Description

Connect and retrieve data from AEMET, SMC and other Spanish meteorological stations services.

Usage

```
get_meteo_from(
  service = c("aemet", "meteocat", "meteoclimatic", "meteogalicia", "ria"),
  options
)
```

Arguments

`service`	Character with the service name (in lower case).
`options`	List with the needed service options. See `services_options` to have more info about the different services and their options.

Details

Depending on the service and the temporal resolution selected, the variables present can change, but all services have at least temperature values.

Value

An sf (spatial) object with the stations meteorological data.

API limits

Some APIs have limits in terms of the data that can be retrieved with one call. For example, AEMET only serves daily data for 31 days in one query. See `vignette('api_limits', package = 'meteospain')` for a detailed explanations of those limits and the ways to retrieve longer periods.

Cache

To avoid unnecessary API calls (especially in rate-limited APIs), results are cached to memory in a `cache_mem` object. This cache is limited to the actual R session and invalidates after 24h. Temporal resolutions below daily are not cached, as they change often.

This cache can be cleared with `clear_meteospain_cache`.

Examples

```
library(meteospain)
library(keyring)

# AEMET (we need a key)
# key_set('aemet')
options_for_aemet <- aemet_options(
  'daily',
  start_date = as.Date('2012-01-16'),
  end_date = as.Date('2012-01-31'),
  api_key = key_get('aemet')
)
get_meteo_from('aemet', options_for_aemet)
```

`get_quota_from` *Get api quota info*

Description

Obtain info about the API quota used

Usage

```
get_quota_from(service = c("meteocat"), options)
```

Arguments

service	Character with the service name (in lower case).
options	List with the needed service options. See services_options to have more info about the different services and their options.

Details

Depending on the service, some APIs allows only a number of data requests. This function access the user quota numbers in the services that allow for this, **(currently only MeteoCat)**

Value

A data frame with the quota info

Examples

```
library(meteospain)
library(keyring)

# MeteoCat (we need a key)
# key_set('meteocat')
api_options <- meteocat_options(api_key = key_get('meteocat'))
get_quota_from('meteocat', api_options)
```

---

get_stations_info_from

Get meteorological stations info

Description

Obtain info and metadata for the available stations in the different services

Usage

```
get_stations_info_from(
  service = c("aemet", "meteocat", "meteoclimatic", "meteogalicia", "ria"),
  options
)
```

Arguments

service	Character with the service name (in lower case).
options	List with the needed service options. See services_options to have more info about the different services and their options.

Details

Depending on the service the metadata available can be different. Also, some services only offer info for active stations (i.e. AEMET), not historical stations, so some mismatch can occur between the stations returned by this function and the stations returned by `get_meteo_from` for historical dates.

Value

An sf (spatial) object with the stations metadata.

Cache

To avoid unnecessary API calls (especially in rate-limited APIs), results are cached to memory in a `cache_mem` object. This cache is limited to the actual R session and invalidates after 24h.

This cache can be cleared with `clear_meteospain_cache`.

Examples

```
library(meteospain)
library(keyring)

# AEMET (we need a key)
# key_set('aemet')
api_options <- aemet_options(api_key = key_get('aemet'))
get_stations_info_from('aemet', api_options)
```

---

get_variables_from *Get variables info*

Description

Obtain information about variables as offered by the APIs

Usage

```
get_variables_from(service = c("meteocat"), options)
```

Arguments

`service`	Character with the service name (in lower case).
`options`	List with the needed service options. See `services_options` to have more info about the different services and their options.

Details

Depending on the service, information about original variable names, menaning, original units... **Currently only MeteoCat is available**, other services will result in an error

Value

A data frame with the variables info

Examples

```
library(meteospain)
library(keyring)

# MeteoCat (we need a key)
# key_set('meteocat')
api_options <- meteocat_options("daily", api_key = key_get('meteocat'))
get_variables_from('meteocat', api_options)
```

`services_options` *Options for meteorological services*

Description

Set the options for accessing the different Spanish meteorological services

Usage

```
aemet_options(
  resolution = c("current_day", "daily", "monthly", "yearly"),
  start_date = Sys.Date(),
  end_date = start_date,
  stations = NULL,
  api_key
)

meteocat_options(
  resolution = c("instant", "hourly", "daily", "monthly", "yearly"),
  start_date = Sys.Date(),
  stations = NULL,
  api_key
)

meteoclimatic_options(resolution = c("current_day"), stations = NULL)

meteogalicia_options(
  resolution = c("instant", "current_day", "daily", "monthly"),
  start_date = Sys.Date(),
  end_date = start_date,
  stations = NULL
)
```

```
ria_options(
  resolution = c("daily", "monthly"),
  start_date = Sys.Date() - 1,
  end_date = start_date,
  stations = NULL
)
```

Arguments

resolution	Character indicating the temporal resolution for the data. Services allows different temporal resolutions that can be present or not in each of them (current_day, instant, daily, monthly).
start_date	Date class object with the start date from which start collecting data. Ignored if resolution is one of current_day or instant.
end_date	Date class object with the end date from which stop collecting data. By default, same date as start_date. Ignored if resolution is one of current_day or instant.
stations	Character vector with the stations codes from which extract data from. If NULL (default) all available stations are returned. See Stations section for more details.
api_key	Character with the API key. NULL by default as not all services require keys. See API Keys section for more details.

Value

A list with the service API options to make the query to obtain the data.

Resolution

Temporal resolutions vary from service to service. Check the "Usage" section to see resolutions available to each service. Possible values are:

- current_day returns the last 12-24h of measures.
- instant returns the last measures available.
- hourly returns the hourly measures.
- daily returns any past date/s with daily aggregation.
- monthly returns any past date/s with monthly aggregation.
- yearly returns any past date/s with yearly aggregation.

Keys

Some services (i.e. AEMET, MeteoCat...) require an API key to access the data. The requirements and process to obtain the key varies from service to service.

- AEMET: Visit https://opendata.aemet.es/centrodedescargas/inicio and follow the instructions at "Obtención de API Key".
- MeteoCat: Visit https://apidocs.meteocat.gencat.cat/ and follow the instructions there.

It is not advisable to use the keys directly in any script shared or publicly available (github...), neither store them in plain text files. One option is using the keyring package for managing and accessing keys.

Stations

Some services accept querying multiple stations at once, and other only allows one station per query:

- AEMET: One or more stations can be provided in a character vector (except for monthly and yearly resolutions, as they only accept one station.
- MeteoCat: One or more stations can be provided in a character vector.
- MeteoGalicia: One or more stations can be provided in a character vector.
- Meteoclimatic: Only one station can be provided. Nevertheless, some codes can be used to retrieve common group of stations: "ES" for all Spanish stations, "ESCAT", "ESCYL", "ESAND"... for the different autonomous communities.
- RIA: API accepts only one station. Nonetheless, an internal loop is performed to retrieve all the stations provided

Examples

```
library(keyring)
library(meteospain)

## AEMET examples -------------------------------------------------------

# setting the key (a prompt will appear in console to supply the API key)
# keyring::key_set(service = 'aemet')

# Options for the last 24h data
current_opts <- aemet_options(
  resolution = 'current_day',
  api = keyring::key_get('aemet')
)

# Options for daily data for January, 1990
daily_opts <- aemet_options(
  resolution = 'daily',
  start_date = as.Date('1990-01-01'),
  end_date = as.Date('1990-01-15'),
  api = keyring::key_get('aemet')
)


## MeteoCat examples ----------------------------------------------------------

# setting the key (a prompt will appear in console to supply the API key)
# keyring::key_set(service = 'meteocat')

# create the options
query_options <- meteocat_options(
```

```
    resolution = 'hourly',
    start_date = as.Date('2020-12-31'),
    api = keyring::key_get('meteocat')
  )


## Meteoclimatic examples -------------------------------------------------

current_opts <- meteoclimatic_options()
# same as before, but more verbose
current_opts <- meteoclimatic_options(resolution = 'current_day', stations = 'ES')


## MeteoGalicia examples --------------------------------------------------

# Options for the last measured data
instant_opts <- meteogalicia_options(resolution = 'instant')

# Options for the last 24h data
current_opts <- meteogalicia_options(resolution = 'current_day')
# same, with stations
current_opts <- meteogalicia_options('current_day', stations = c('10045', '10046'))

# Options for daily data for January, 2000
daily_opts <- meteogalicia_options(
  resolution = 'daily',
  start_date = as.Date('2000-01-01'),
  end_date = as.Date('2000-01-31')
)

# Options for monthly data for year 2000
monthly_opts <- meteogalicia_options(
  resolution = 'monthly',
  start_date = as.Date('2000-01-01'),
  end_date = as.Date('2000-12-31')
)


  library(keyring)
  library(meteospain)

  ## RIA examples ----------------------------------------------------------

  # Options for daily data for April, 2020
  daily_opts <- ria_options(
    resolution = 'daily',
    start_date = as.Date('2020-04-01'),
    end_date = as.Date('2020-04-30')
  )
```

Index