Package 'frostr'

May 8, 2026

Type Package

Title R API to MET Norway's 'Frost' API

Version 0.2.0

Description An R API to MET Norway's 'Frost' API <https://frost.met.no/index.html> to retrieve data as data frames. The 'Frost' API, and the underlying data, is made available by the Norwegian Meteorological Institute (MET Norway). The data and products are distributed under the Norwegian License for Open Data 2.0 (NLOD) <https://data.norge.no/nlod/en/2.0> and Creative Commons 4.0 <https://creativecommons.org/licenses/by/4.0/>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports httr, jsonlite, tibble, tidyr

Suggests testthat

NeedsCompilation no

Author Iman Ghayoornia [aut, cre]

Maintainer Iman Ghayoornia <ghayoornia.iman@gmail.com>

Repository CRAN

Date/Publication 2020-08-03 10:40:09 UTC

Contents

get_available_qualitycodes *Get metadata on existing quality flags in the "observation" resource*

Description

`get_available_qualitycodes()` returns a data frame with quality flags that describes the quality of an observation. The function requires input for `client_id`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_available_qualitycodes(client_id,
                           fields = NULL,
                           language = NULL,
                           return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`fields`	A character vector. The field to include in the response (i.e. output). If this parameter is set, then only the specified field is returned as a data frame. If not set, then all fields will be returned in the response as a list. The options are `NULL` (default), `"summarized"` and `"details"`.
`language`	A string. The language of the fields in the response. The options are "en-US" (default), "nb-NO" (Norwegian, Bokmål), and "nn-NO" (Norwegian, Nynorsk).
`return_response`	A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame of quality flags, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get metadata for quality codes
qualitycodes <- get_available_qualitycodes(client_id = frost_client_id)

# Get the summarized metadata for quality codes
summarized_df <- get_available_qualitycodes(client_id = frost_client_id,
```

```
                                        fields = "summarized")

## End(Not run)
```

`get_available_timeseries` *Get metadata on available time series in the "observation" resource*

Description

`get_available_timeseries()` retrieves metadata on available time series that you can get from the Frost API "observations" resource with `get_observations()`. The function requires input for `client_id` and `sources`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_available_timeseries(client_id,
                         sources,
                         reference_time = NULL,
                         elements = NULL,
                         time_offsets = NULL,
                         time_resolutions = NULL,
                         time_series_ids = NULL,
                         performance_categories = NULL,
                         exposure_categories = NULL,
                         levels = NULL,
                         level_types = NULL,
                         level_units = NULL,
                         fields = NULL,
                         return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`sources`	A character vector. The station IDs of the data sources to get observations for. For example, "SN18700" is the station ID for "Blindern". The full list of station IDs can be retrieved with `get_sources()`.
`reference_time`	A string. The time range to get observations for in either extended ISO-8601 format or the single word "latest".
`elements`	A character vector. The elements to get observations for. The full list of elements can be retrieved with the `get_elements()`.
`time_offsets`	A character vector. The time offsets to get observations for provided as a vector of ISO-8601 periods, e.g. `c("PT6H", "PT18H")`.

`time_resolutions`
: A character vector. The time resolutions to get observations for provided as a vector of ISO-8601 periods e.g. `c("PT6H", "PT18H")`.

`time_series_ids`
: A numeric vector. The internal time series IDs to get observations for as a vector of integers, e.g. c(0, 1).

`performance_categories`
: A character vector. The performance categories to get observations for as a vector of letters, e.g. `c("A", "C")`.

`exposure_categories`
: A numeric vector. The exposure categories to get observations for as a vector of integers, e.g. `c(1, 2)`.

`levels`
: A numeric vector. The sensor levels to get observations for as a vector of integers, e.g. `c(1, 2, 10, 20)`.

`level_types`
: A character vector. The sensor level types to get data for.

`level_units`
: A character vector. The sensor level units to get data for.

`fields`
: A character vector. Fields to include in the response (i.e. output). If this parameter is specified, then only these fields are returned in the response. If not specified, then all fields will be returned in the response.

`return_response`
: A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame with metadata about available time series, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get metadata on available time series for Blindern station (with station
# ID SN18700)
sources <- "SN18700"

obs_available_ts <- get_available_timeseries(client_id = frost_client_id,
                                             sources   = sources)

## End(Not run)
```

get_elements *Get metadata about the weather and climate elements that are defined in the Frost API*

Description

`get_elements()` retrieves metadata about weather and climate elements defined for use in the Frost API. The function requires an input for `client_id`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_elements(client_id,
             ids = NULL,
             names = NULL,
             descriptions = NULL,
             units = NULL,
             code_tables= NULL,
             statuses = NULL,
             calculation_method = NULL,
             categories = NULL,
             time_offsets = NULL,
             sensor_levels = NULL,
             old_element_codes = NULL,
             old_units = NULL,
             cf_standard_names = NULL,
             cf_cell_methods = NULL,
             cf_units = NULL,
             cf_versions = NULL,
             fields = NULL,
             language = NULL,
             return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`ids`	A character vector. The element IDs to get metadata for.
`names`	A character vector. The element names to get metadata for.
`descriptions`	A character vector. The descriptions to get metadata for.
`units`	A character vector. The units to get metadata for.
`code_tables`	A character vector. The code tables to get metadata for.
`statuses`	A character vector. The statuses to get metadata for.
`calculation_method`	A string. The calculation method as a JSON filter. Supports the following keys: baseNames, methods, innerMethods, periods, innerPeriods, thresholds, methodDescriptions, innerMethodDescriptions, methodUnits, and innerMethodUnits.

`categories` A character vector. The categories to get metadata for.

`time_offsets` A character vector. The time offsets to get metadata for.

`sensor_levels` A string. The sensor levels to get metadata for as a JSON filter. Supports the following keys: levelTypes, units, defaultValues, and values.

`old_element_codes`
A character vector. The old MET Norway element codes to get metadata for.

`old_units` A character vector. The old MET Norway units to get metadata for.

`cf_standard_names`
A character vector. The CF standard names to get metadata for.

`cf_cell_methods`
A character vector. The CF cell methods to get metadata for.

`cf_units` A character vector. The CF units to get metadata for.

`cf_versions` A character vector. The CF versions to get metadata for.

`fields` A character vector. Fields to include in the response (i.e. output). If this parameter is specified, then only these fields are returned in the response. If not specified, then all fields will be returned in the response.

`language` A string. The language of the fields in the response. The options are "en-US" (default), "nb-NO" (Norwegian, Bokmål), and "nn-NO" (Norwegian, Nynorsk).

`return_response`
A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame with metadata about climate and weather elements, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get data for all elements
elements_df <- get_elements(client_id = frost_client_id)

## End(Not run)
```

get_element_codetables *Get metadata about code tables for the "elements" resource*

Description

`get_element_codetables()` retrieves metadata about code tables. A code table defines a small number of discrete values for an element. The function requires input for `client_id`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_element_codetables(client_id,
                       ids = NULL,
                       fields = NULL,
                       language = NULL,
                       return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`ids`	A character vector. The element IDs to get metadata for.
`fields`	A character vector. The field to include in the response (i.e. output). If this parameter is set, then only the specified field is returned as a data frame. If not set, then all fields will be returned in the response as a list. The options are "summarized" and "details".
`language`	A string. The language of the fields in the response. The options are "en-US" (default), "nb-NO" (Norwegian, Bokmål), and "nn-NO" (Norwegian, Nynorsk).
`return_response`	A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame with metadata about code tables, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get the full code table
code_tables <- get_element_codetables(client_id = frost_client_id)
```

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

`get_locations` *Get metadata for the location names defined in the Frost API*

Description

`get_locations()` retrieves metadata about location names defined for use in the Frost API. The function requires an input for `client_id`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_locations(client_id,
              names = NULL,
              geometry = NULL,
              fields = NULL,
              return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`names`	A character vector. The location names that you want metadata for.
`geometry`	A string. Get Frost API location names defined by a specified geometry. Geometries are specified as either "nearest(POINT(...))" or "POLYGON(...)" using well-known text representation for geometry (WKT).
`fields`	A character vector. Fields to include in the response (i.e. output). If this parameter is specified, then only these fields are returned in the response. If not specified, then all fields will be returned in the response.
`return_response`	A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame with metadata about location names, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get all location names
locations_df <- get_locations(client_id = frost_client_id)
```

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

`get_observations` *Get weather observations from the "observation" resource in the Frost API*

Description

`get_observations()` retrieves historical weather data from the Frost API. This is the core resource for retrieving actual observation data from MET Norway's data storage systems. The function requires input for `client_id`, `sources`, `reference_time`, and `elements`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

Usage

```
get_observations(client_id,
                 sources,
                 reference_time,
                 elements,
                 maxage = NULL,
                 limit = NULL,
                 time_offsets = NULL,
                 time_resolutions = NULL,
                 time_series_ids = NULL,
                 performance_categories = NULL,
                 exposure_categories = NULL,
                 qualities = NULL,
                 levels = NULL,
                 include_extra = NULL,
                 fields = NULL,
                 return_response = FALSE)
```

Arguments

`client_id`	A string. The client ID to use to send requests to the Frost API.
`sources`	A character vector. The station IDs of the data sources to get observations for. For example, "SN18700" is the station ID for "Blindern". The full list of station IDs can be retrieved with `get_sources()`.
`reference_time`	A string. The time range to get observations for in either extended ISO-8601 format or the single word "latest".
`elements`	A character vector. The elements to get observations for. The full list of elements can be retrieved with the `get_elements()`.

`maxage` A string. The maximum observation age as an ISO-8601 period, e.g. `"P1D"`. This parameter is only applicable when `reference_time = "latest"`. Defaults to "PT3H".

`limit` A string or a positive integer. The maximum number of observation times to be returned for each combination of source and element, counting from the most recent time. This parameter is only applicable when `reference_time = "latest"`. Set `limit = "all"` to get all available times or a positive integer. Defaults to 1.

`time_offsets` A character vector. The time offsets to get observations for provided as a vector of ISO-8601 periods, e.g. `c("PT6H", "PT18H")`.

`time_resolutions`
A character vector. The time resolutions to get observations for provided as a vector of ISO-8601 periods e.g. `c("PT6H", "PT18H")`.

`time_series_ids`
A numeric vector. The internal time series IDs to get observations for as a vector of integers, e.g. c(0, 1).

`performance_categories`
A character vector. The performance categories to get observations for as a vector of letters, e.g. `c("A", "C")`.

`exposure_categories`
A numeric vector. The exposure categories to get observations for as a vector of integers, e.g. `c(1, 2)`.

`qualities` A numeric vector. The qualities to get observations for as a vector of integers, e.g. `c(1, 2)`.

`levels` A numeric vector. The sensor levels to get observations for as a vector of integers, e.g. `c(1, 2, 10, 20)`.

`include_extra` An integer. If this parameter is set to 1, and extra data is available, then this data is included in the response. Extra data currently consists of the original observation value and the 16-character control info.

`fields` A character vector. Fields to include in the response (i.e. output). If this parameter is specified, then only these fields are returned in the response. If not specified, then all fields will be returned in the response.

`return_response`
A logical. If set to `TRUE`, then the function returns the response from the GET request. If set to `FALSE` (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame of historical weather observations, or the response of the GET request, depending on the boolean value set for `return_response`.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"
```

```
# Get daily data for temperature, rain, and wind speed for 2018
sources <- "SN18700"
reference_time <- "2018-01-01/2018-12-31"
elements <- c("mean(air_temperature P1D)",
              "sum(precipitation_amount P1D)",
              "mean(wind_speed P1D)")

observations_df <- get_observations(client_id      = frost_client_id,
                                    sources        = sources,
                                    reference_time = reference_time,
                                    elements       = elements)

## End(Not run)
```

`get_sources` *Get metadata for source entities defined in the Frost API*

Description

`get_sources()` retrieves metadata about (data) source entities defined for use in the Frost API. The function requires an input for `client_id`. The other function arguments are optional, and default to `NULL`, which means that the response from the API is not filtered on these parameters.

NB: At the time of writing (2019-06-08), the Frost API "sources" resource only returns country names in Norwegian.

Usage

```
get_sources(client_id,
            ids = NULL,
            types = NULL,
            geometry = NULL,
            nearest_max_count = NULL,
            valid_time = NULL,
            name = NULL,
            country = NULL,
            county = NULL,
            municipality = NULL,
            wmo_id = NULL,
            station_holder = NULL,
            external_ids = NULL,
            icao_code = NULL,
            ship_code = NULL,
            wigos_id = NULL,
            fields = NULL,
            return_response = FALSE)
```

Arguments

client_id	A string. The client ID to use to send requests to the Frost API.
ids	A character vector. The Frost API source ID(s) that you want metadata for.
types	A string. The type of Frost API source that you want metadata for. Must be set to either "SensorSystem", "InterpolatedDataset", or "RegionDataset". Defaults to NULL, which returns all three types.
geometry	A string. Get Frost API sources defined by a specified geometry. Geometries are specified as either "nearest(POINT(...))" or "POLYGON(...)" using well-known text representation for geometry (WKT).
nearest_max_count	A string. The maximum number of sources returned when using "nearest(POINT(...))" for the geometry argument. Defaults to 1.
valid_time	A string. The time interval for which the sources have been, or still are, valid (or applicable). Specify as "<date>/<date>" or "<date>/now" where <date> is a date in the ISO-8601 format (YYYY-MM-DD). Defaults to "now" if not specified, which returns only currently valid sources.
name	A string. The data source name to get metadata for.
country	A string. The country name or country code to get metadata for.
county	A string. The county name or county ID to get metadata for.
municipality	A string. The municipality to get metadata for.
wmo_id	A string. The WMO ID to get metadata for.
station_holder	A string. The station holder name to get metadata for.
external_ids	A character vector. The external ID to get metadata for.
icao_code	A string. The ICAO code to get metadata for.
ship_code	A string. The ship code to get metadata for.
wigos_id	A string. The WIGOS ID to get metadata for.
fields	A character vector. Fields to include in the response (i.e. output). If this parameter is specified, then only these fields are returned in the response. If not specified, then all fields will be returned in the response.
return_response	A logical. If set to TRUE, then the function returns the response from the GET request. If set to FALSE (default), then the function returns a tibble (data frame) of the content in the response.

Value

The function returns either a data frame with metadata about source entities, or the response of the GET request, depending on the boolean value set for return_response.

Examples

```
## Not run:
frost_client_id <- "<YOUR FROST CLIENT ID>"

# Get data for all sources
sources <- get_sources(client_id = frost_client_id)

# Get data for sources in Norway
sources_norway <- get_sources(client_id = frost_client_id,
                              country   = "NO")

## End(Not run)
```

Index