

Package: wateRinfo (via r-universe)

October 29, 2024

Title Download Time Series Data from Waterinfo.be

Version 0.3.0.9065

Description wateRinfo facilitates access to waterinfo.be (<<https://www.waterinfo.be>>), a website managed by the Flanders Environment Agency (VMM) and Flanders Hydraulics Research. The website provides access to real-time water and weather related environmental variables for Flanders (Belgium), such as rainfall, air pressure, discharge, and water level. The package provides functions to search for stations and variables, and download time series.

License MIT + file LICENSE

URL <https://github.com/ropensci/wateRinfo>,
<https://docs.ropensci.org/wateRinfo>

BugReports <https://github.com/ropensci/wateRinfo/issues>

Depends R (>= 2.10)

Imports dplyr, httr, jsonlite, openssl, lubridate (>= 1.6.0), rlang,
utils

Suggests covr, ggplot2, knitr, rmarkdown, testthat

LazyData true

Encoding UTF-8

VignetteBuilder knitr

RoxygenNote 7.1.2

Repository <https://ropensci.r-universe.dev>

RemoteUrl <https://github.com/ropensci/wateRinfo>

RemoteRef main

RemoteSha 4511f52e315877078a38a4354613a2223e0c917a

Contents

air_pressure	2
check_period_format	3
get_stations	3
get_timeseries_tsid	5
get_token	6
get_variables	8
is_supported_variable	9
liedekerke	9
print.waterinfo_api	10
resolve_datasource	10
resolve_timeseriesgroupid	11
supported_frequencies	12
supported_variables	12
Index	13

air_pressure	<i>Air pressure data of January 1st, 2017</i>
--------------	---

Description

A dataset compiled by downloading 1 day of air pressure data for the available stations of Waterinfo.be

Usage

```
air_pressure
```

Format

A data frame with 710 rows and 13 variables:

ts_id identifier of the downloaded time serie
Timestamp datetime
Value measured value of the variable
Quality Code Quality code of the measurement
station_latitude latitude coordinate
station_longitude longitude coordinate
station_id identifier of the measurement station
station_no short code name of the measurement station
station_name full name of the measurement station
stationparameter_name parameter name on station level
parameter_type_name parameter type name
ts_unitsymbol unit of the variable
dataprovider provider of the time series value

Source

<https://www.waterinfo.be/>

check_period_format *Check period string format*

Description

Check if the format of the period is conform the specifications of VMM

Usage

```
check_period_format(period_string)
```

Arguments

period_string input string according to format required by waterinfo: The period string is provided as P#Y#M#DT#H#M#S, with P defines 'Period', each # is an integer value and the codes define the number of... Y - years M - months D - days T required if information about sub-day resolution is present H - hours D - days M - minutes S - seconds Instead of D (days), the usage of W - weeks is possible as well Examples of valid period strings: P3D, P1Y, P1DT12H, PT6H, P1Y6M3DT4H20M30S.

Value

str period string itself if valid

Examples

```
check_period_format("P2DT6H") # period of 2 days and 6 hours
check_period_format("P3D") # period of 3 days
```

get_stations *Get list of stations for a variable*

Description

For a given timeseriesgroup (variable), provide a list of measurement stations providing data. An overview of the variables is provided by the function [supported_variables](#).

Usage

```
get_stations(variable_name = NULL, frequency = "15min", token = NULL)
```

Arguments

variable_name	char valid nam of available variable as timeseriesgroup
frequency	char valid frequency for the given variable, for most variables, the 15min frequency is available
token	token to use with the call (optional, can be retrieved via get_token)

Format

A data.frame with 10 variables:

ts_id Unique timeseries identifier to access time series data corresponding to a combination of the station, measured variable and frequency.

station_latitude Latitude coordinates of the station (WGS84)

station_longitude Longitude coordinates of the station (WGS84)

station_id Identifier of the station as used in the waterinfo backend

station_no Station ID as provided on the waterinfo.be website.

station_name Official name of the measurement station.

stationparameter_name Station specific variable description.

parametername_name Measured variable description.

ts_unitsymbol Unit of the variable.

dataproducer Data provider of the time series data.

The URL of the specific request is provided as a comment attribute to the returned data.frame. Use `comment(df)` to get the request URL.

Details

For the moment, this only works for measurement stations of VMM (meetnet 1), and stations from other measurement data sources are not included in the list

Value

data.frame with an overview of the available stations for the requested variable

See Also

`supported_variables`

Examples

```
get_stations('irradiance')
get_stations('soil_saturation')
```

get_timeseries_tsid *Download timeseries data from waterinfo.be*

Description

Using the ts_id codes and by providing a given date period, download the corresponding time series from the waterinfo.be website

Usage

```
get_timeseries_tsid(
  ts_id,
  period = NULL,
  from = NULL,
  to = NULL,
  datasource = 1,
  token = NULL
)
```

Arguments

ts_id	waterinfo.be database ts_id, defining a timeserie variable and frequency it is defined.
period	input string according to format required by waterinfo: De period string is provided as P#Y#M#DT#H#M#S, with P defines 'Period', each # is an integer value and the codes define the number of... Y - years M - months D - days T required if information about sub-day resolution is present H - hours D - days M - minutes S - seconds Instead of D (days), the usage of W - weeks is possible as well Examples of valid period strings: P3D, P1Y, P1DT12H, PT6H, P1Y6M3DT4H20M30S.
from	date of datestring as start of the time series
to	date of datestring as end of the time series
datasource	int [0-4] defines the 'meetnet' of which the measurement station is part of. VMM based stations are net '1', MOW-HIC is net '2'
token	token to use with the call (optional, can be retrieved via get_token)

Format

A data.frame with 3 variables:

Timestamp Datetime of the measurement.

Value Measured value.

Quality Code Quality code of the measurement, dependent on the data source used:

- VMM Quality Code Interpretation (datasource 1)
 - 10/110 - Excellent

- 30/100/130 - Good
- 50/150 - Moderate
- 70/170 - Poor
- 80/180 - Estimated
- 90/190 - Suspect
- 220 - Default
- -1 - Missing
- HIC Quality Code Interpretation (datasource 2)
 - 40 - Good
 - 80 - Estimated
 - 120 - Suspect
 - 200 - Unchecked
 - 60 - Complete
 - 160 - Incomplete
 - -1 - Missing
- Aggregated timeseries
 - 40 - Good
 - 100 - Estimated
 - 120 - Suspect
 - 200 - Unchecked
 - -1 - Missing

The URL of the specific request is provided as a comment attribute to the returned data.frame. Use `comment(df)` to get the request URL.

Value

data.frame with the timestamps, values and quality code

Examples

```
get_timeseries_tsid("35055042", from = "2017-01-01", to = "2017-01-02")
get_timeseries_tsid("5156042", period = "P3D")
get_timeseries_tsid("55419010", from = "2017-06-01", to = "2017-06-03",
                    datasource = 4)
```

get_token

Get waterinfo Token

Description

Retrieve a fresh waterinfo token. A token is not required to get started, see Details section for more information.

Usage

```
get_token(
  client = NULL,
  client_id = NULL,
  client_secret = NULL,
  token_url = "http://download.waterinfo.be/kiwis-auth/token"
)
```

```
is.expired(token)
```

```
expires.in(token)
```

Arguments

client	base64 encoded client containing the client id and client secret, separated by :
client_id	client id string
client_secret	client secret string
token_url	url to get the token from
token	a token object

Details

Notice you do not need to get a token right away to download data. For limited and irregular downloads, a token will not be required. The amount of data downloaded from waterinfo.be is limited via a credit system. When you require more extended data requests, request a download token.

Either client or client_id and client_secret need to be passed as arguments. If provided, client is always used. Tokens remain valid for 24 hours, after which a fresh one must be acquired. To limit load on the server, token objects should be reused as much as possible until expiration in stead of creating fresh ones for each call.

The client_id and client_secret provided in the examples are for test purposes, get your very own client via <hydrometrie@waterinfo.be>.

Value

An object of class token containing the token string with the token_url, token type and moment of expiration as attributes.

Examples

```
# Get token via client_id and client_secret
client_id <- '32dcece-826c-4b98-9c2d-b16978f0ca6a'
client_secret <- '4ada871a-9528-4b4d-bfd5-b570ce8f4d2d'
my_token <- get_token(client_id = client_id, client_secret = client_secret)
print(my_token)

# get token via client
client <- paste0('MzJKY2V1Y2UtODI2Yy00Yjk4LTljMmQtYjE2OTc4ZjBjYTZh0jRhZGE4',
```

```

                                'NzFhLTk1MjgtNGI0ZC1iZmQ1LWI1NzBjZThmNGQyZA==')
my_token <- get_token(client = client)
print(my_token)
is.expired(my_token)
expires.in

# Use the token when requesting for data (i.e. get_* functions), e.g.
get_stations(variable_name = "verdamping_monteith", token = my_token)

```

get_variables

Get list of variables for a given station

Description

Get list of variables for a given station

Usage

```
get_variables(station_no, token = NULL)
```

Arguments

station_no	'stations-nummer' as it appears on the download page of waterinfo.be
token	token to use with the call (optional, can be retrieved via get_token)

Format

A data.frame with 6 variables:

station_name Official name of the measurement station.

station_no Station ID as provided on the waterinfo.be website.

ts_id Unique timeseries identifier to access time series data corresponding to a combination of the station, measured variable and frequency.

ts_name Timeseries identifier description name as provided by 'waterinfo.be'.

parametername Measured variable description.

stationparameter_name Station specific variable description.

The URL of the specific request is provided as a comment attribute to the returned data.frame. Use `comment(df)` to get the request URL.

Value

data.frame with the station_name, station_no, ts_id, ts_name and parametername for each of the variables for this station.

Examples

```
variables_overpelt <- get_variables("ME11_002")
```

is_supported_variable *Check if variable is supported by VMM ts group id*

Description

Check if variable is supported by VMM ts group id

Usage

```
is_supported_variable(variable_name)
```

Arguments

variable_name char

Value

Raise error when variable is not supported directly, otherwise NULL

Examples

```
is_supported_variable("wind_speed")
```

liedekerke *Soil moisture data of Liedekerke, January 2017*

Description

A dataset compiled by downloading 1 day of soil moisture data for the Liedekerke measurement station of Waterinfo.be

Usage

```
liedekerke
```

Format

A data frame with 23,816 rows and 9 variables:

ts_id identifier of the downloaded time serie

Timestamp datetime

Value measured value of the variable

Quality Code Quality code of the measurement

station_name full name of the measurement station

station_no short code name of the measurement station

ts_name type/frequency of the time serie
parameter_type_name parameter type name
stationparameter_name parameter name on station level

Source

<https://www.waterinfo.be/>

`print.waterinfo_api` *Custom print function of the API request response*

Description

Custom print function of the API request response

Usage

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

Arguments

<code>x</code>	<code>waterinfo_api</code>
<code>...</code>	args further arguments passed to or from other methods.

`resolve_datasource` *Define the datasource using the station number*

Description

Using the 'stations-nummer' as provided on waterinfo.be, this function tries to identify the data-source to use for the particular variable

Usage

```
resolve_datasource(station_no)
```

Arguments

<code>station_no</code>	'stations-nummer' as it appears on the download page of waterinfo.be
-------------------------	--

Details

Notice that VMM did not provide this in the official documentation, but this has just been derived by checking the API response as such. A more automated and less hard-coded approach would be beneficial, but this data is not available at the moment.

Value

integer 1 for VMM, 4 for other 'meetnetten' (HIC,...)

Examples

```
resolve_datasource('akl03e-1066')
resolve_datasource('K07_0M421')
```

resolve_timeseriesgroupid

Get timeseriesgroupID for a supported variable

Description

Translate the usage of available variables to the corresponding timeseriesgroupID, based on the provided lookup table from VMM

Usage

```
resolve_timeseriesgroupid(variable_name, frequency = "15min")
```

Arguments

variable_name valid variable name, supported by VMM API
frequency valid frequency for the given variable

Details

Remark that this information is NOT based on a query, but on information provided by the package itself to make variable names more readable

The lookup table is provided as external data of the package, see inst/extdata

Value

list containing the timeseriesgroup_id of the variable frequency combination

Examples

```
resolve_timeseriesgroupid("rainfall", "15min")
```

supported_frequencies *VMM supported timeseriesgroups frequencies*

Description

Provide list of VMM supported frequencies for a given timeseriesgroupID in either dutch or english

Usage

```
supported_frequencies(variable_name)
```

Arguments

variable_name char name of a valid variable in either dutch or english

Examples

```
supported_frequencies('rainfall')
```

supported_variables *VMM supported timeseriesgroups variables*

Description

Provide list of VMM supported variables in the timeseriesgroupID in either dutch or english

Usage

```
supported_variables(language = "nl")
```

Arguments

language char nl (dutch) or en (english) variable names

Value

data.frame containing the variable names in either english or dutch

Examples

```
# Request supported variables in Dutch
supported_variables("nl")

# Request supported variables in English
supported_variables("en")
```

Index

* datasets

- air_pressure, 2
- liedekerke, 9

air_pressure, 2

check_period_format, 3

expires.in(get_token), 6

get_stations, 3

get_timeseries_tsid, 5

get_token, 4, 5, 6, 8

get_variables, 8

is.expired(get_token), 6

is_supported_variable, 9

liedekerke, 9

print.token(get_token), 6

print.waterinfo_api, 10

resolve_datasource, 10

resolve_timeseriesgroupid, 11

show.token(get_token), 6

supported_frequencies, 12

supported_variables, 3, 12

token(get_token), 6