Package: noaaoceans (via r-universe)

August 25, 2024

Type Package

Title Collect Ocean Data from NOAA

Version 0.3.0.9000

Maintainer Sean Warlick <warlick.sean@gmail.com>

Description Provides a small set of tools for collecting data from National Oceanic and Atmospheric Administration (NOAA) data sources. The functions provided in the package are wrappers around NOAA's existing APIs which is found at <https://api.tidesandcurrents.noaa.gov/api/prod/>.

URL www.github.com/warlicks/noaaoceans
www.warlicks.github.io/noaaoceans

BugReports https://github.com/warlicks/noaaoceans/issues

License MIT + file LICENSE

Encoding UTF-8

Suggests testthat, covr, knitr, rmarkdown, dplyr, httptest, ggplot2, maps, mapdata

RoxygenNote 7.1.1

Imports httr, jsonlite, rvest, xml2

VignetteBuilder knitr

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

RemoteUrl https://github.com/warlicks/noaaoceans

RemoteRef HEAD

RemoteSha 7178349a2e0e226df96a56c36426e3e350844288

Contents

`coops_station_inventory` *Gather Co-OPS Station Data History*

Description

Find the historical data availability for a CO-OPS station. This data is obtained by scraping the data inventory page for the station. See the page for the#' Barbuda(station_id=9761115) as an example.

Usage

```
coops_station_inventory(station_id)
```

Arguments

`station_id`	is a character string that provides the a 7 character station id.

Details

In the returned data frame each row represents a particular oceanographic or meteorological measurement. The name of the measurement is provided in the first column (**Type**). The second column, **From**, provides a timestamp indicating the earliest available data for the measurement. The third column, **To**, provides the last date and time when the measurement is available. When there are gaps in availability there will be two rows from a given measurement. See the table below as an example.

Type	**From**	**To**
Wind	2011-06-10 21:06	2019-11-30 06:36
Air Temperature	2011-06-10 20:48	2019-11-30 06:36
Water Temperature	2011-06-10 20:48	2013-03-10 03:48
Water Temperature	2015-04-03 13:06	2019-11-30 06:36

As of the release of version 0.20.0 there are thirteen stations where the data inventory is missing and an **Error** will be returned if queried with `coops_station_inventory()`. The list of stations without a data inventory can be accessed by calling `noaaoceans:::known_missing_inventory()`. Due to heavy use of JavaScript on data inventory pages and a desire to keep package dependencies to a minimum the list of known stations has been hard coded. Please create an Issue or Pull Request to update the list stations missing data inventory.

Value

A data frame.

Examples

```
# Working station to show results.
inventory_df<- coops_station_inventory(station_id=9761115)
print(inventory_df)

# Station with known missing data inventory
coops_station_inventory(station_id=8517986)
```

`list_coops_stations` *Find All NOAA Stations*

Description

This function produces a data frame with all NOAA stations. The list of stations is retrieved from NOAA's website when the function is called.

Usage

```
list_coops_stations()
```

Details

In the returned data frame there is one row for each station. The name, location and date that the station was established are included as columns. In addition, there are columns that provide the status of various sensors at the station is included. The column names indicate the type of sensor

In the status columns a value of *1* indicates that sensor is working A *0* indicates that the sensor is not working. If a particular station does not have the capability indicated by the column name, the value provided is `NA`

Value

A data frame.

Examples

```
# Do Not Run

station_df <- list_coops_stations()
```

`noaaoceans` *noaaoceans: A package for collecting oceans and weather data from NOAA.*

Description

The noaaoceans package provides several functions to access NOAA APIs. It includes functions to access the CO-OPS API and metadata for each of the tide sensor stations.

`query_coops_data` *Retrieve Tides Data From NOAA CO-OPS API*

Description

Retrieve Tides Data From NOAA CO-OPS API

Usage

```
query_coops_data(
  station_id,
  start_date,
  end_date,
  data_product,
  units = "english",
  time_zone = "gmt",
  datum = NULL,
  interval = NULL,
  bin = NULL
)
```

Arguments

`station_id` is a character string that provides the a 7 character station id.

`start_date` is a character string that specifies the start date for the retrieval period. Dates can be specified in the following formats: *yyyyMMdd*, *yyyyMMdd HH:mm*, *MM/dd/yyyy*, or *MM/dd/yyyy HH:mm*.

`end_date` is a character string that specifies the end date for the retrieval period. Dates can be specified in the following formats: *yyyyMMdd*, *yyyyMMdd HH:mm*, *MM/dd/yyyy*, or *MM/dd/yyyy HH:mm*.

`data_product` specifies the data product to be returned. See CO-OPS API Documentation for the available data products.

`units` a character string specifying if the data should be returned using metric or English units. Defaults to `'english'`.

`time_zone` a character string specifying what time zone information the data should be returned with. Options include Greenwich Mean Time `'gmt'`, Local Standard Time `'lst'`, and Local Standard/Local Daylight Time `'lst_ldt'`. Local times refer to the local time of the specified station. The default is `'gmt'`

`datum` a character string indicating the datum that should be returned. See CO-OPS API Documentation for the available datums.

`interval` a character string that specifies the interval for which Meteorological data is returned. The API defaults to every six minutes and does not need to be specified. Other option include hourly `'h'` and `'hilo'`. The retrieval time period specified by **start_date** and **end_date** to create restrictions on the intervals that can be returned. See CO-OPS API Documentation for details

`bin` the bin number for the indicated currents station. If a bin is not specified for a PORTS station, the data is returned using a predefined real-time bin.

Value

a data frame.

Examples

```
# Do Not Run
a <- query_coops_data('9414290',
                      '20170101',
                      '20170201',
                      'predictions',
                       interval = 'hilo',
                       datum = 'MLLW')
```

`query_derived_products` *Query Data From NOAA's CO-OPS Derived Product API*

Description

Provides access to data available from NOAA's CO-OPS Derived Product API. Four derived data products are available through the API: 1) Top Ten Water Levels, 2) Annual Flood Days, 3) Extreme Water Levels and 4) Sea Level Trends. More detail about each data product is available with the API's documentation

Usage

```
query_derived_products(
  station_id = NULL,
  product_name = NULL,
  year = NULL,
  affil = NULL,
  units = "english"
)
```

Arguments

`station_id` an optional string that provides the a 7 character station id. If omitted the derived product API returns data for all stations.

`product_name` a string providing the name of the derived data product. Derived products include Top Ten Water Levels (`'toptenwaterlevels'`), Annual Flood Days (`'annualflooddays'`), Extreme Water Levels (`'extremewaterlevels'`) and Sea Level Trends (`'sealeveltrends'`).

`year` an optional string used to limit the results from the annual flood days product to the indicated year. The argument is ignored if used with other data products.

`affil` an optional argument used to limit the results from the Sea Level Trends product to U.S. (`'US'`) or Global stations (`'Global'`). If omitted with Sea Level Trends all stations are returned. The argument is ignored if used with other data products.

`units` a character string specifying if the data should be returned using metric or English units. Defaults to `'english'`

Details

Results are returned as a data frame specific to each product. For the variables returned with each endpoint see the appropriate section in the documentation.

- Top Ten Water Levels
- Annual Flood Days
- Extreme Water Levels
- Sea Level Trends

Value

a data frame. See the documentation for the specific product for the details an it's content.

Examples

```
# Not Run
query_derived_products(station_id = "1611400",
                       product_name = "sealeveltrends")
```

`query_metadata` *Query CO-OPS API for Station Metadata*

Description

Provides easy access to the CO-OPS Metadata API. The api makes information about measurement stations available to users. Information about a single station or a collection of stations can be accessed. Depending on the type of station queried different information is returned.

Usage

```
query_metadata(
  station_id = NULL,
  resource = NULL,
  type = NULL,
  ports = NULL,
  units = "english",
  radius = NULL,
  bin = NULL
)
```

Arguments

`station_id` an optional string that provides the a 7 character station id. If omitted the derived product API returns data for all stations.

`resource` a character string indicating they type of information to request for a specific station. A list of resource identifiers is available in the API Documentation

`type` a character string indicating the sensor of interest. Specifying a sensor of interest returns a data frame with all stations that have the particular sensor. A list of sensor identifiers is available in the API Documentation

`ports` A two character string indicating specific ports.

`units` a character string specifying if the data should be returned using metric or English units. Defaults to `'english'`

`radius` an optional numeric argument indicating the radius in nautical miles to search for nearby stations

`bin` an optional (positive integer) argument to requests for currents station harmonic constituents. If not specified, all the bins will be returned.

Value

A data frame. The content of the data frame is dependent on the API call. See the API documentation for specifics.

Examples

```
# Query a single stations sensors.
sensor_df <- query_metadata('9414290', 'sensors')

# Query all stations
all_stations_df <- query_metadata()
```

Index