Package ‘cimir’

February 17, 2021

Title Interface to the CIMIS Web API
Version 0.4-1
Description Connect to the California Irrigation Management Information System (CIMIS) Web API. See the CIMIS main page <https://cimis.water.ca.gov> and web API documentation <https://et.water.ca.gov> for more information.
License GPL (>= 3)
URL https://github.com/mkoohafkan/cimir
BugReports https://github.com/mkoohafkan/cimir/issues
Depends R (>= 3.4)
Imports curl (>= 4.3), glue (>= 1.3), stringr (>= 1.4), dplyr (>= 0.8), tidyr (>= 1.0), jsonlite (>= 1.6), purrr (>= 0.3), rlang (>= 0.4)
Encoding UTF-8
LazyData true
Suggests knitr (>= 1.21), rmarkdown (>= 1.11)
VignetteBuilder knitr
RoxygenNote 7.1.1
NeedsCompilation no
Author Michael Koohafkan [aut, cre]
Maintainer Michael Koohafkan <michael.koohafkan@gmail.com>
Repository CRAN
Date/Publication 2021-02-17 22:40:03 UTC

R topics documented:
cimir .......................................................... 2
cimis_compass_to_degrees .................................. 3
cimis_data .................................................... 3
This package provides an R interface to the California Irrigation Management Information System (CIMIS) Web API. In order to use this package, you will need to create a CIMIS account and request a web services AppKey.

Package options

cimir uses the following `options()` to configure behavior:

- `cimir.appkey`: The CIMIS AppKey to use for queries.
- `cimir.timeout`: The maximum time to wait for a response from the CIMIS Web API.

Alternatively, the CIMIS App Key can be saved to an environment variable `CIMIS_APPKEY`.

Author(s)

Maintainer: Michael Koohafkan <michael.koohafkan@gmail.com>

See Also

Useful links:

- https://github.com/mkoohafkan/cimir
- Report bugs at https://github.com/mkoohafkan/cimir/issues
cimis_compass_to_degrees

Compass Direction To Degrees

Description

Convert the Compass direction labels to degrees.

Usage

cimis_compass_to_degrees(x)

Arguments

x  A vector of compass directions, i.e. the data item labels "DayWindNnw", "Day-WindSse", etc. Recognized directions are North-northeast (NNE), East-northeast (ENE), East-southeast (ESE), South-southeast (SSE), South-southwest (SSW), West-southwest (WSW), West-northwest (WNW), and North-northwest (NNW).

Value

A numeric vector of degrees corresponding to the middle azimuth of the corresponding compass direction.

See Also

cimis_degrees_to_compass()

Examples

cimis_compass_to_degrees("day-wind-nne")
cimis_compass_to_degrees(c("SSE", "SSW", "wsw", "Wnw", "nnw"))

cimis_data

Query CIMIS Data

Description

Query CIMIS data using the Web API.
Usage

cimis_data(
  targets,  # geographies or weather stations of interest. This parameter may specify one or many stations, zip codes, coordinates, or street addresses; however, you are not allowed to mix values from different categories. This means the targets parameter must contain only stations, only zip codes, only coordinates, or only street addresses. You will receive an error if you attempt to mix different category types. The formats are accepted:
  - A comma delimited list of WSN station numbers
  - A comma delimited list of California zip codes
  - A semicolon delimited list of decimal - degree coordinates
  - A semicolon delimited list of street addresses
  start.date,  # Specifies the start date. The data format is "yyyy-mm-dd".
  end.date,  # Specifies the end date. The data format is "yyyy-mm-dd".
  items,  # specifies one or more comma-delimited data elements to include in your response. See data_items() for a complete list of possible data element values. Default: day-asce-eto, day-precip, day-sol-rad-avg, day-vap-pres-avg, day-air-tmp-max, day-air-tmp-min, day-air-tmp-avg, day-rel-hum-max, day-rel-hum-min, day-rel-hum-avg, day-dew-pnt, day-wind-spd-avg, day-wind-run, day-soil-tmp-avg.
  measure.unit = c("E", "M"),  # The unit of measure may be either "E" for English units or "M" for metric units. The value of this parameter will affect data values in the response. For example, designating English units will result in temperature values being returned in Fahrenheit rather than Celsius.
  prioritize.SCS = TRUE
)

Arguments

targets  # geographies or weather stations of interest. This parameter may specify one or many stations, zip codes, coordinates, or street addresses; however, you are not allowed to mix values from different categories. This means the targets parameter must contain only stations, only zip codes, only coordinates, or only street addresses. You will receive an error if you attempt to mix different category types. The formats are accepted:
  - A comma delimited list of WSN station numbers
  - A comma delimited list of California zip codes
  - A semicolon delimited list of decimal - degree coordinates
  - A semicolon delimited list of street addresses

start.date  # Specifies the start date. The data format is "yyyy-mm-dd".
end.date  # Specifies the end date. The data format is "yyyy-mm-dd".
items  # specifies one or more comma-delimited data elements to include in your response. See data_items() for a complete list of possible data element values. Default: day-asce-eto, day-precip, day-sol-rad-avg, day-vap-pres-avg, day-air-tmp-max, day-air-tmp-min, day-air-tmp-avg, day-rel-hum-max, day-rel-hum-min, day-rel-hum-avg, day-dew-pnt, day-wind-spd-avg, day-wind-run, day-soil-tmp-avg.
measure.unit  # The unit of measure may be either "E" for English units or "M" for metric units. The value of this parameter will affect data values in the response. For example, designating English units will result in temperature values being returned in Fahrenheit rather than Celsius.
prioritize.SCS  # This parameter is relevant only when the targets parameter contains zip code(s). If TRUE, the Spatial CIMIS System (SCS) will be used as the preferred data provider.

Value

A tibble object.

Examples

if(is_key_set()) {
cimis_data(targets = 170, start.date = Sys.Date() - 4,  
  end.date = Sys.Date() - 1)
cimis_degrees_to_compass

 Degrees to Compass Direction

Description

Convert decimal degrees to Compass direction.

Usage

cimis_degrees_to_compass(x)

Arguments

x

A vector of directions in decimal degrees.

Details

Degrees are labeled with their corresponding Primary InterCardinal compass direction, following the convention of the CIMIS daily wind data items.

Value

A factor vector of compass directions.

See Also

cimis_compass_to_degrees()

Examples

cimis_degrees_to_compass(c(30, 83, 120, 140, 190, 240, 300, 330))
cimis_degrees_to_compass(cimis_compass_to_degrees(c("NNE", "ENE", "ESE", "SSE", "SSW", "WSW", "WNW", "NNW"))))
### cimis_flags

**CIMIS Data Flags**

**Description**

List CIMIS data quality control flags.

**Usage**

```r
cimis_flags(type = c("Severe", "Informative"), period = "Current")
```

**Arguments**

- `type` The type of data flag, i.e. "Severe" or "Informative".
- `period` The Time period that data was collected, i.e., "Current" or "Former" (pre-1995).

**Value**

a dataframe of data flags.

**See Also**

[CIMIS Data Overview - Quality Control](#)

**Examples**

```r
cimis_flags()
cimis_flags("Informative")
cimis_flags("Severe", period = "Former")
```

---

### cimis_format_location

**Format CIMIS Station Location**

**Description**

Format the latitude and longitude of station in Decimal Degrees (DD) or Hour Minutes Seconds (HMS).

**Usage**

```r
cimis_format_location(d, format = c("DD", "HMS"))
```
cimis_items

Arguments

d A data frame of CIMIS data results.

format The format to use, either Decimal Degrees ("DD") or Hour Minutes Seconds ("HMS").

Value

The data frame, with a new "Latitude" and "Longitude" columns replacing the "HmsLatitude" and "HmsLongitude" columns.

Examples

```r
if(is_key_set()) {
  d = cimis_station(170)
  cimis_format_location(d, "DD")
  cimis_format_location(d, "HMS")
}
```

cimis_items

CIMIS Data Items

Description

List CIMIS data items.

Usage

```r
cimis_items(type = c("Daily", "Hourly"))
```

Arguments

| type          | The type of data item, i.e. "Daily" or "Hourly". |

Value

a dataframe of data items.

Examples

```r
cimis_items()
```
cimis_split_query \hspace{1cm} Split CIMIS Query

Description
Split a large CIMIS query into multiple smaller queries based on a time interval.

Usage
cimis_split_query(targets, start.date, end.date, items, max.records = 1750L)

Arguments
targets \hspace{1cm} geographies or weather stations of interest. This parameter may specify one or many stations, zip codes, coordinates, or street addresses; however, you are not allowed to mix values from different categories. This means the targets parameter must contain only stations, only zip codes, only coordinates, or only street addresses. You will receive an error if you attempt to mix different category types. The formats are accepted:
- A comma delimited list of WSN station numbers
- A comma delimited list of California zip codes
- A semicolon delimited list of decimal - degree coordinates
- A semicolon delimited list of street addresses

start.date \hspace{1cm} Specifies the start date. The data format is "yyyy-mm-dd".

end.date \hspace{1cm} Specifies the end date. The data format is "yyyy-mm-dd".

items \hspace{1cm} specifies one or more comma-delimited data elements to include in your response. See data_items() for a complete list of possible data element values. Default: day-asce-eto, day-precip, day-sol-rad-avg, day-vap-pres-avg, day-air-tmp-max, day-air-tmp-min, day-air-tmp-avg, day-rel-hum-max, day-rel-hum-min, day-rel-hum-avg, day-dew-pnt, day-wind-spd-avg, day-wind-run, day-soil-tmp-avg.

max.records \hspace{1cm} The maximum number of records returned by a query. The default value is the maximum data limit allowed by the CIMIS Web API (1,750 records).

Details
Queries are not split by targets or items, i.e. each resulting query will include all targets and items.

Value
A data frame with columns "targets", "start.date", "end.date", and "items".
cimis_station

Examples

```r
cimis_split_query(170, "2000-01-01", "2010-12-31", "day-air-tmp-avg")
cimis_split_query(c(149, 170), "2018-01-01", "2018-12-31",
c("day-air-tmp-avg", "hly-air-tmp", "hly-rel-hum"))
```

---

**Description**

Query CIMIS station metadata.

**Usage**

```r
cimis_station(station)
cimis_spatial_zipcode(zipcode)
cimis_zipcode(zipcode)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>station</td>
<td>The station ID. If missing, metadata for all stations is returned.</td>
</tr>
<tr>
<td>zipcode</td>
<td>The (spatial) zip code. If missing, metadata for all stations is returned.</td>
</tr>
</tbody>
</table>

**Value**

A tibble object.

**Examples**

```r
if(is_key_set()) {
  cimis_station()
  cimis_station()
  cimis_zipcode()
  cimis_spatial_zipcode()
}
```
**cimis_to_datetime**  
*To Datetime*

**Description**
Collapse The Date and Hour columns to a single DateTime Column.

**Usage**
```r
cimis_to_datetime(d)
```

**Arguments**
- `d`  
  A data frame of CIMIS data results.

**Details**
According to the CIMIS Report FAQs, all CIMIS data is based on Pacific Standard Time (PST).

**Value**
The data frame, with a new "Datetime" column replacing the "Date" and "Hour" columns.

**Examples**
```r
if(is_key_set()) {
  d = cimis_data(targets = 170, start.date = Sys.Date() - 4,
                 end.date = Sys.Date() - 1, items = "hly-air-tmp")
  cimis_to_datetime(d)
}
```

---

**set_key**  
*Specify CIMIS API key*

**Description**
Enter your CIMIS AppKey for web API data access.

**Usage**
```r
set_key(key = NULL)
```

**remove_key()**

**is_key_set()**
set_key

Arguments

key A CIMIS AppKey.

Examples

```r
## Not run:
set_key("YOUR-APP-KEY")
is_key_set()
remove_key()

## End(Not run)
```
Index

cimir, 2

cimir-package (cimir), 2

cimis_compass_to_degrees, 3

cimis_compass_to_degrees(), 5

cimis_data, 3

cimis_degrees_to_compass, 5

cimis_degrees_to_compass(), 3

cimis_flags, 6

cimis_format_location, 6

cimis_items, 7

cimis_spatial_zipcode (cimis_station), 9

cimis_split_query, 8

cimis_station, 9

cimis_to_datetime, 10

cimis_zipcode (cimis_station), 9

is_key_set (set_key), 10

options(), 2

remove_key (set_key), 10

set_key, 10