Package 'weatherapi'

February 18, 2022

Type Package

Title Package to wrap the Weather API

Version 0.1.0

Author c(person(given = ``Evelyn'',
    family = ``Sugihermanto'',
    role = c(``aut'', ``cre''),
    email = ``evelynsugihermanto@yahoo.com''),
    person(given = ``Anqi'',
    family = ``Li'',
    role = c(``aut'', ``cre''),
    email = ``lianqi20209@gmail.com''),
    person(given = ``Xin'',
    family = ``Tian'',
    role = c(``aut'',``cre''),
    email = ``tjyh111@163.com''),
    person(given = ``Val'',
    family = ``Veeramani'',
    role = c(``aut'',``cre''),
    email = ``veeramanival@gmail.com'')

Maintainer Evelyn Sugihermanto <evelynsugihermanto@yahoo.com>

Description The weatherapi package is a wrapper package for the Weather API.

License MIT + file LICENSE

Encoding UTF-8

Imports httr, jsonlite

RoxygenNote 7.1.2

Suggests rmarkdown, knitr, testthat, devtools, ggplot2, reshape2,
    tidyverse, maps

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

---

api_key   *Get or set API_KEY value*

Description

This package will need a Weather API key residing in the environment variable `API_KEY`. Set it in the `.Renviron` file in the home directory. See vignette for more details on how to set up the API key.

Usage

```
api_key(force = FALSE)
```

Arguments

force   A boolean to force set new API key for current environment. To force set a new API key set it to TRUE, the default value is FALSE.

Value

A vector containing the API key

Examples

```
# api_key() # To display API Key that was set in the .Renviron file
# api_key(force=TRUE) # To change the API key, see vignette for details
```

---

data_request   *get the astronomy data of specific date time and city from weather-api.com*

Description

get the astronomy data of specific date time and city from weatherapi.com

Usage

```
data_request(query)
```

Arguments

query   A list contain information of city, date, and api_key, represented by q, dt, key, repectively.

Value

The http get response containing astronomy information.

Examples

```
data_request(list(key="abc", q="London", dt="2021-01-01"))
```

---

get_astronomy	*A wrapper function to obtain the the astronomy data of a the desired city and date time from the weatherapi.com.*

Description

A wrapper function to obtain the the astronomy data of a the desired city and date time from the weatherapi.com.

Usage

```
get_astronomy(city, date)
```

Arguments

city	A string indicating city (e.g., "London", "Beijing").
date	A string of the form yyyy-mm-dd indicating date (e.g., "2022-02-10").

Value

A dataframe with the columns 'sunrise', 'sunset', 'moonrise', 'moonset', 'moon_phase', and 'moon_illumination'.

Examples

```
get_astronomy("Beijing", "2022-01-10")
```

---

get_current_weather	*Get current weather information*

Description

Get the current real-time weather and air-quality information

Usage

```
get_current_weather(location, air_quality = "yes")
```

Arguments

location	A vector of location
air_quality	A string of "yes" or "no" to show air quality information. Default is show air quality information ("yes").

Value

A data frame of current weather information (see vignette for details)

Examples

```
get_current_weather("Kelowna")
get_current_weather(c("Kelowna", "Vancouver"), "no")
```

---

get_history_astro_information
                              Get history astronomical information for a specific day

---

Description

Get history astronomical information for a specific day

Usage

```
get_history_astro_information(q, dt)
```

Arguments

q	A string containing the location, which can be US Zipcode, UK Postcode, Canada Postalcode, IP address, Latitude/Longitude (decimal degree) or city name.
dt	A string containing the date for query, which should be within the last 7 days.

Value

A data frame containing the astronomical information for the day.

Examples

```
get_history_astro_information("London","2022-02-12")
```

---

get_history_daily_weather
                              Get history weather information for a specific day

---

Description

Get history weather information for a specific day

Usage

```
get_history_daily_weather(q, dt)
```

Arguments

q	A string containing the location, which can be US Zipcode, UK Postcode, Canada Postalcode, IP address, Latitude/Longitude (decimal degree) or city name.
dt	A string containing the date for query, which should be within the last 7 days.

Value

A data frame containing the daily weather information for the day.

Examples

```
get_history_daily_weather("London","2022-02-12")
```

---

get_history_hourly_weather

Get history hourly weather information for a specific hour in a specific day

---

Description

Get history hourly weather information for a specific hour in a specific day

Usage

```
get_history_hourly_weather(q, dt, h)
```

Arguments

q	A string containing the location, which can be US Zipcode, UK Postcode, Canada Postalcode, IP address, Latitude/Longitude (decimal degree) or city name.
dt	A string containing the date for query, which should be within the last 7 days.
h	An integer containing the hour for query, which be in the range of 0 to 23.

Value

A data frame containing the daily weather information for the hour in the day.

Examples

```
get_history_hourly_weather("London","2022-02-12",4)
```

---

get_sports_events   *Get sports information*

Description

Get a list of all upcoming sports events in a location

Usage

```
get_sports_events(location, sport)
```

Arguments

location	A vector of location
sport	A string of sport type ("football","cricket","golf")

Value

A data frame of upcoming sports events in a location (see vignette for details)

Examples

```
get_sports_events("London", "football")
get_sports_events(c("London", "Oxford"), "football")
```

---

get_time_zone *Get time zone information*

Description

Get the local time-zone information for a location

Usage

```
get_time_zone(location)
```

Arguments

location   A vector of location

Value

A data frame of local time-zone information (see vignette for details)

Examples

```
get_time_zone("Kelowna")
get_time_zone(c("Kelowna", "Vancouver"))
```