Package: scrappy (via r-universe)

September 13, 2024

Title A Simple Web Scraper

Version 0.0.2

Description A group of functions to scrape data from different websites, for academic purposes.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL https://github.com/villegar/scrappy/, https://villegar.github.io/scrappy/, https://scrappy.robertovillegas-diaz.com/

BugReports https://github.com/villegar/scrappy/issues/

Language en-GB

Imports dplyr, httr, jsonlite, lubridate, magrittr, purrr, rvest, stringr, tibble, xml2

Depends R (>= 2.10)

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

RemoteUrl https://github.com/villegar/scrappy

RemoteRef HEAD

RemoteSha 0fb4491f0367c19fde0787ffd3e0491a6a94b26d

Contents

digimap_os 2
duration2datetime 3
find_a_gp 4
google_maps 5
newa3_stations 6

newa_nrcc 7
newa_nrcc3 8
newa_stations 9
print 10
wait_to_load 10

Index **11**

`digimap_os` *(Assisted) request of EDINA's Digimap data from the Ordnance Survey*

Description

Digimap's Ordnance Survey collection provides a full range of topographic Ordnance Survey data for the UK. Note that this function helps you to request the data, once your request has been processed, you will have to manually download the data (email instructions will be provided by Digimap).

Usage

```
digimap_os(
  client,
  area_name = NULL,
  dataset = NULL,
  format = NULL,
  version = NULL,
  org = Sys.getenv("ORG"),
  sleep = 1
)
```

Arguments

`client`	`RSelenium` client.
`area_name`	String with UK national grid name (e.g., 'SD', 'SD20'). See ordnancesurvey.co.uk/documents/resources/guide-to-nationalgrid.pdf.
`dataset`	String with the name of the data set to download (e.g., 'NTM' for National Tree Map or 'Terrain-5 DTM' for the OS Terrain 5 Digital Terrain Model). See https://digimap.edina.ac.uk/help/our-maps-and-data/os_products/.
`format`	String with the data format (e.g., 'SHAPE'). See https://digimap.edina.ac.uk/help/our-maps-and-data/os_products/.
`version`	String with the version of the data set (e.g., 'July 2023'). See https://digimap.edina.ac.uk/help/our-maps-and-data/os_products/.
`org`	String with your organisation name (for login purposes, this can be done manually). Done only once per session.
`sleep`	Integer with number of seconds to use as pause between actions on the web page.

Value

Logic value with the status of the data request.

Source

https://digimap.edina.ac.uk/os

`duration2datetime` *Convert duration to date-time Convert a string with a duration (e.g. 'an hour ago') to a date-time string, based on a reference time*

Description

Convert duration to date-time Convert a string with a duration (e.g. 'an hour ago') to a date-time string, based on a reference time

Usage

```
duration2datetime(
  str,
  ref_time = Sys.time(),
  output_format = "%Y-%m-%d %H:%M:%S %Z"
)
```

Arguments

`str`	String with a duration (see examples)
`ref_time`	Reference time (default: `Sys.time()`, current time)
`output_format`	String with the format of the output (default: `"%Y-%m-%d %H:%M:%S %Z"`)

Value

Date-time object based on the input string, `str`, and the reference time `ref_time`.

Examples

```
duration2datetime("a minute ago")
duration2datetime("an hour ago")
duration2datetime("a day ago")
duration2datetime("a week ago")
duration2datetime("a month ago")
duration2datetime("a year ago")
duration2datetime("2 minutes ago")
duration2datetime("2 hours ago")
duration2datetime("2 days ago")
duration2datetime("2 weeks ago")
duration2datetime("2 months ago")
duration2datetime("2 years ago")
```

`find_a_gp` *Scrape GP practices*

Description

Scrape GP practices near a given postcode

Usage

```
find_a_gp(
  client,
  postcode,
  base = "https://www.nhs.uk/service-search/find-a-gp",
  sleep = 1
)
```

Arguments

`client`	`RSelenium` client.
`postcode`	String with the target postcode.
`base`	String with the base URL for Google Maps website.
`sleep`	Integer with number of seconds to use as pause between actions on the web page.

Value

Data frame with GP practices near the given `postcode`.

Examples

```
## Not run:
# Create RSelenium session
rD <- RSelenium::rsDriver(browser = "firefox", port = 4544L, verbose = FALSE)

# Retrieve GP practices near L69 3GL
# (Waterhouse building, University of Liverpool)
wh_gps_tb <- scrappy::find_a_gp(rD$client, postcode = "L69 3GL")

# Stop server
rD$server$stop()

## End(Not run)
```

google_maps *Scrape Google Maps' reviews*

Description

Scrape Google Maps' reviews

Usage

```
google_maps(
  client,
  name,
  place_id = NULL,
  base = "https://www.google.com/maps/search/?api=1&query=",
  sleep = 1,
  max_reviews = 100,
  result_id = 1,
  with_text = FALSE
)
```

Arguments

client	RSelenium client.
name	String with the name of the target place.
place_id	String with the unique ID of the target place, useful when more than one place has the same name.
base	String with the base URL for Google Maps website.
sleep	Integer with number of seconds to use as pause between actions on the web page.
max_reviews	Integer with the maximum number of reviews to scrape. The number of existing reviews will define the actual number of reviews returned.
result_id	Integer with the result position to use, only relevant when multiple matches for the given name are found.
with_text	Boolean value to indicate if the max_reviews should only account for those reviews with a comment.

Value

Tibble with the reviews extracted from Google Maps.

Examples

```
## Not run:
# Create RSelenium session
rD <- RSelenium::rsDriver(browser = "firefox", port = 4544L, verbose = FALSE)
```

```
# Retrieve reviews for Sefton Park in Liverpool
sefton_park_reviews_tb <-
  scrappy::google_maps(
    client = rD$client,
    name = "Sefton Park",
    place_id = "ChIJrTCHJVkge0gRm1LWF0fSPgw",
    max_reviews = 20
  )

sefton_park_reviews_tb_with_text <-
  scrappy::google_maps(
    client = rD$client,
    name = "Sefton Park",
    place_id = "ChIJrTCHJVkge0gRm1LWF0fSPgw",
    max_reviews = 20,
    with_text = TRUE
  )
# Stop server
rD$server$stop()

## End(Not run)
```

`newa3_stations` *NEWA v3 Weather Stations dataset*

Description

A dataset containing information of 801 weather stations in the Network for Environment and Weather Applications (NEWA) version 3 at Cornell University.

Usage

```
data(newa3_stations)
```

Format

A data frame with 801 rows and 10 variables:

name Station's name.

state State where the station is located.

code Station's code.

affiliation Entity to which the entity is affiliated.

affiliation_url Entity's URL.

latitude Station's latitude.

longitude Station's longitude.

elevation Station's elevation.

start_year Start year (data available).

is_icao Boolean flag to indicate if the station is part of the International Civil Aviation Organization (ICAO) (e.g is an airport).

Author(s)

Network for Environment and Weather Applications <support@newa.zendesk.com>

Source

https://newa.cornell.edu

`newa_nrcc` *Retrieve data from NEWA at Cornell University*

Description

Retrieve Weather data from the Network for Environment and Weather Applications (NEWA) at Cornell University.

Usage

```
newa_nrcc(
  client,
  year,
  month,
  station,
  base = "http://newa.nrcc.cornell.edu/newaLister",
  interval = "hly",
  sleep = 6,
  table_id = "#dtable",
  path = getwd(),
  save_file = TRUE
)
```

Arguments

`client`	`RSelenium` client.
`year`	Numeric value with the year.
`month`	Numeric value with the month.
`station`	String with the station abbreviation. Check the http://newa.cornell.edu/index.php?page=station-pages for a list.
`base`	Base URL (default: http://newa.nrcc.cornell.edu/newaLister).
`interval`	String with data interval (default: hly, hourly).
`sleep`	Numeric value with the number of seconds to wait for the page to load the results (default: 6 seconds).
`table_id`	String with the unique HTML ID assigned to the table containing the data (default: `#dtable`)
`path`	String with path to location where CSV files should be stored (default: `getwd()`).
`save_file`	Boolean flag to indicate whether or not the output should be stored as a CSV file.

Value

Tibble with the data retrieved from the server.

Examples

```
## Not run:
# Create RSelenium session
rD <- RSelenium::rsDriver(browser = "firefox", port = 4544L, verbose = FALSE)
# Retrieve data for the Geneva (Bejo) station on 2020/12
scrappy::newa_nrcc(rD$client, 2020, 12, "gbe")
# Stop server
rD$server$stop()

## End(Not run)
```

`newa_nrcc3` *Retrieve data from NEWA v3.0 at Cornell University*

Description

Retrieve Weather data from the Network for Environment and Weather Applications (NEWA) version 3.0 at Cornell University.

Usage

```
newa_nrcc3(
  year,
  month,
  day,
  hour,
  station,
  base = "https://hrly.nrcc.cornell.edu/stnHrly"
)
```

Arguments

`year`	Numeric value with the start year.
`month`	Numeric value with the start month.
`day`	Numeric value with the start day.
`hour`	Numeric value with the start hour.
`station`	String with the station abbreviation. Check `scrappy::newa3_stations` for a list of stations and abbreviations.
`base`	Base URL (default: `https://hrly.nrcc.cornell.edu/stnHrly`).

Value

List of data frames with `hourly`, `daily`, `hourly_forecast`, and daily forecast (`daily_forecast`) data.

Examples

```
scrappy::newa_nrcc3(2021, 12, 01, 00, "gbe")
```

`newa_stations` *NEWA Weather Stations dataset*

Description

A dataset containing information of 718 weather stations in the Network for Environment and Weather Applications (NEWA) at Cornell University.

Usage

```
data(newa_stations)
```

Format

A data frame with 718 rows and 3 variables:

name Station's name.

state State where the station is located.

code Station's code.

Author(s)

Network for Environment and Weather Applications <support@newa.zendesk.com>

Source

http://newa.cornell.edu/index.php?page=station-pages

print *Print Values*

Description

Print Values

Print Google Maps' reviews

Usage

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

Arguments

x	an object used to select a method.
...	further arguments passed to or from other methods.

wait_to_load *Wait until page has finished loading*

Description

Wait until page has finished loading the element with the tag value

Usage

```
wait_to_load(client, using = "css", value = "body", sleep = 1)
```

Arguments

client	RSelenium client.
using	String with the property to use to find the element (e.g. "css", "xpath", etc.) (default: "css").
value	String with the tag of the page element to wait to load (default: "body").
sleep	Numeric value with the number of seconds to wait for the page to load the results (default: 1 second).

Index

∗ **datasets**
    newa3_stations, 6
    newa_stations, 9

digimap_os, 2
duration2datetime, 3

find_a_gp, 4

google_maps, 5

newa3_stations, 6
newa_nrcc, 7
newa_nrcc3, 8
newa_stations, 9

print, 10

wait_to_load, 10