Package ‘ralger’

April 12, 2020

Type Package
Title Easy Web Scraping
Version 2.0.0
Author Mohamed El Fodil Ihaddaden
Maintainer Mohamed El Fodil Ihaddaden <ihaddaden.fodeil@gmail.com>
Description The goal of ‘ralger’ is to facilitate web scraping in R.
   The user has the ability to extract a vector with scrap(), a tidy dataframe using tidy_scrap(), a table with table_scrap() and web links with weblink_scrap().
License MIT + file LICENSE
Encoding UTF-8
LazyData true
URL https://github.com/feddelegrand7/ralger
BugReports https://github.com/feddelegrand7/ralger/issues
VignetteBuilder knitr
Imports magrittr, rvest, xml2, testthat, tidyr, dplyr, stringr,
   robotstxt, crayon, purrr
Suggests knitr, rmarkdown
RoxygenNote 7.1.0
NeedsCompilation no
Repository CRAN
Date/Publication 2020-04-12 07:40:03 UTC

R topics documented:

  paragraphs_scrap ...................................................... 2
  scrap ................................................................. 2
  table_scrap .......................................................... 3
  tidy_scrap ............................................................ 4
  titles_scrap ........................................................... 5
  weblink_scrap .......................................................... 5
Index

paragraphs_scrap     Website text paragraph scraping

Description
This function is used to scrape text paragraphs from a website.

Usage
paragraphs_scrap(link, contain = NULL, collapse = FALSE, askRobot = FALSE)

Arguments
- link: the link of the web page to scrape
- contain: filter the paragraphs according to the character string provided.
- collapse: if TRUE the paragraphs will be collapsed into one element and the contain argument ignored.
- askRobot: logical. Should the function ask the robots.txt if we’re allowed or not to scrap the web page? Default is FALSE.

Value
a character vector.

Examples

# Extracting the paragraphs displayed on the health topic of the New York Times

link <- "https://www.nytimes.com/section/health"

paragraphs_scrap(link)

scrap     Simple website scraping

Description
This function is used to scrape one element from a website.

Usage
scrap(link, node, clean = FALSE, askRobot = FALSE)
**table_scrap**

**Arguments**
- **link** the link of the web page to scrape
- **node** the HTML or CSS element to consider, the SelectorGadget tool is highly recommended
- **clean** logical. Should the function clean the extracted vector or not? Default is FALSE.
- **askRobot** logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE.

**Value**
a character vector

**Examples**

```r
# Extracting imdb top 250 movie titles

link <- "https://www.imdb.com/chart/top/
node <- ".titleColumn a"

scrap(link, node)
```

---

**table_scrap**  
**HTML table scraping**

**Description**

This function is used to scrape an html table from a website.

**Usage**

table_scrap(link, choose = 1, header = T, askRobot = FALSE)

**Arguments**
- **link** the link of the web page containing the table to scrape
- **choose** an integer indicating which table to scrape
- **header** do you want the first line to be the leader (default to TRUE)
- **askRobot** logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE.

**Value**
a data frame object.
Examples

# Extracting premier ligue 2019/2020 top scorers

```r
link <- "https://www.topscorersfootball.com/premier-league"
table_scrap(link)
```

<table>
<thead>
<tr>
<th>tidy_scrap</th>
<th>Website Tidy scraping</th>
</tr>
</thead>
</table>

Description

This function is used to scrape a tibble from a website.

Usage

```r
tidy_scrap(link, nodes, colnames, clean = FALSE, askRobot = FALSE)
```

Arguments

- `link`: the link of the web page to scrape
- `nodes`: the vector of HTML or CSS elements to consider, the SelectorGadget tool is highly recommended.
- `colnames`: the names of the expected columns.
- `clean`: logical. Should the function clean the extracted tibble or not? Default is FALSE.
- `askRobot`: logical. Should the function ask the robots.txt if we're allowed or not to scrape the web page? Default is FALSE.

Value

A tidy data frame.

Examples

# Extracting imdb movie titles and rating

```r
link <- "https://www.imdb.com/chart/top/
my_nodes <- c(".titleColumn a", "strong")
names <- c("title", "rating")
tidy_scrap(link, my_nodes, names)
```
titles_scrap

Website title scraping

Description

This function is used to scrape titles from a website. Useful for scraping daily electronic newspapers’ titles.

Usage

titles_scrap(link, contain = NULL, askRobot = FALSE)

Arguments

- `link`: the link of the web page to scrape
- `contain`: filter the titles according to the character string provided.
- `askRobot`: logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE

Value

a character vector

Examples

# Extracting the current titles of the New York Times
link <- "https://www.nytimes.com/

titles_scrap(link)

weblink_scrap

Website web links scraping

Description

This function is used to scrape web links from a website.

Usage

weblink_scrap(link, contain = NULL, askRobot = FALSE)
weblink_scrap

Arguments

- **link**: the link of the web page to scrape
- **contain**: filter the web links according to the character string provided.
- **askRobot**: logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE.

Value

- a character vector.

Examples

```r
# Extracting the web links within the World Bank research and publications page

link <- "https://www.worldbank.org/en/research"

weblink_scrap(link)
```
Index

paragraphs_scrap, 2
scrap, 2

table_scrap, 3
tidy_scrap, 4
titles_scrap, 5
weblink_scrap, 5