Package ‘ralger’

October 14, 2022

Type Package
Title Easy Web Scraping
Version 2.2.4
Maintainer Mohamed El Fodil Ihaddaden <ihaddaden.fodeil@gmail.com>
Description The goal of ‘ralger’ is to facilitate web scraping in R.
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 rvest, xml2, tidyr, dplyr, stringr, robotstxt, crayon, curl, stringi
Suggests knitr, testthat, rmarkdown, covr
RoxygenNote 7.1.1
NeedsCompilation no
Author Mohamed El Fodil Ihaddaden [aut, cre], Ezekiel Ogundepo [ctb], Romain François [ctb]
Repository CRAN
Date/Publication 2021-03-17 23:10:02 UTC

R topics documented:

attribute_scrap .................................................. 2
images_noalt_scrap ............................................ 3
images_preview .................................................. 3
images_scrap .................................................... 4
paragraphs_scrap ................................................. 5
**Attribute Scrap**

**Scraping attributes from HTML elements**

**Description**

This function is used to scrape attributes from HTML elements.

**Usage**

```r
attribute_scrap(link, node, attr, askRobot = FALSE)
```

**Arguments**

- `link`: the link of the web page to scrape.
- `node`: the HTML element to consider.
- `attr`: the attribute to scrape.
- `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://ropensci.org/

# scraping the class attributes' names from all the anchor
attribute_scrap(link = link, node = "a", attr = "class")
```
images_noalt_scrap  
Scrape Images URLS that don’t have 'alt' attributes

Description
Scrape Images URLS that don’t have 'alt' attributes

Usage
images_noalt_scrap(link, askRobot = FALSE)

Arguments
- link: the URL of the web page
- 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 of images’ URL without "alt" attribute

Examples

images_noalt_scrap(link = "https://www.r-consortium.org/")

images_preview  
Scrape Images URLs

Description
Scrape Images URLs

Usage
images_preview(link, askRobot = FALSE)

Arguments
- link: the link of the web page
- askRobot: logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE.
Value
Images URLs

Examples

images_preview(link = "https://rstudio.com/")

---

images_scrap Scrape Images from a Web Page

Description
Scrape Images from a Web Page

Usage

images_scrap(link, imgpath = getwd(), extn, askRobot = FALSE)

Arguments

- **link**: the link of the web page
- **imgpath**: the path of the images. Defaults to the current directory
- **extn**: the extension of the image: png, jpeg ...
- **askRobot**: logical. Should the function ask the robots.txt if we’re allowed or not to scrape the web page? Default is FALSE.

Value
Images

Examples

## Not run:
images_scrap(link = "https://rstudio.com/", extn = "png")

## End(Not run)
Description

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

Usage

```r
paragraphs_scrap(
  link,
  contain = NULL,
  case_sensitive = FALSE,
  collapse = FALSE,
  askRobot = FALSE
)
```

Arguments

- `link`: the link of the web page to scrape
- `contain`: filter the paragraphs according to the character string provided.
- `case_sensitive`: logical. Should the contain argument be case sensitive? defaults to FALSE
- `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

```r
# Extracting the paragraphs displayed on the health page 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**

```r
scrap(link, node, clean = FALSE, askRobot = FALSE)
```

**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.
**tidy_scrap**

**Usage**

```
table_scrap(link, choose = 1, header = TRUE, fill = FALSE, 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)
- `fill`: logical. Should be set to TRUE when the table has an inconsistent number of columns.
- `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
link <- "https://www.topscorersfootball.com/premier-league"
table_scrap(link)
```

---

**tidy_scrap**

**Website Tidy scraping**

**Description**

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

**Usage**

```
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

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

tidy_scrap(link, my_nodes, names)
```

## Description

This function is used to scrape titles (h1, h2 & h3 html tags) from a website. Useful for scraping daily electronic newspapers’ titles.

## Usage

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

## Arguments

- `link`: the link of the web page to scrape
- `contain`: filter the titles according to a character string provided.
- `case_sensitive`: logical. Should the contain argument be case sensitive? defaults to 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

# Extracting the current titles of the New York Times

```
link <- "https://www.nytimes.com/

titles_scrap(link)
```
Description

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

Usage

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

Arguments

- link: the link of the web page to scrape
- contain: filter the web links according to the character string provided.
- case_sensitive: logical. Should the contain argument be case sensitive? defaults to 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

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

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

weblink_scrap(link)
Index

attribute_scrap, 2
images_noalt_scrap, 3
images_preview, 3
images_scrap, 4
paragraphs_scrap, 5
scrap, 6
table_scrap, 6
tidy_scrap, 7	
titles_scrap, 8
weblink_scrap, 9