

Package ‘seoR’

January 29, 2018

Type Package

Title SEO Related Analyses

Version 0.1.0

Author Daniel Schmeh

Maintainer Daniel Schmeh <danielschmeh@gmail.com>

Description Help SEOs (Search Engine Optimization) retrieve relevant informations from various APIs or websites.

It is possible to scrape SEO-relevant parts of a website. So you are able to extract links, meta-tags, h-tags, and many more.

The package also provides functions to scrape informations form search engines. Like indexed pages, number of results for a given keyword or complete search results.

The third part of the package are the SEO-Tool APIs, that are connected. It's possible to get Informations from 'Whois', 'Google Pagespeed' and many more direct in R.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

Imports jsonlite, httr, rvest, stringr, XML, robotstxt, igrph, RCurl, dplyr, readr, digest, xml2, utils

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-29 11:42:15 UTC

R topics documented:

allowedByRobots	2
choice	3
domainAge	3
downloadSitemap	4
extractLinks	4
getBingResults	5

googleSuggest	5
host_tld	6
hrefLang	6
htag	7
htag_count	7
htag_input	8
HTMLcanonical	8
HTMLdescription	9
HTMLdescription_length	9
HTMLrobots	10
HTMLtitle	10
HTMLtitle_length	11
internal_all_file	11
isIndexable	12
keywordResults	12
lastCached	13
linkCount	13
mobileFriendly	14
mozLinkMetrics	14
mozUrlMetrics	15
pagesInIndex	16
pageSpeed	17
responseCode	17
screamingfrog_crawlVsSitemap	18
screamingfrog_internalPagerank	18
seoDiver	19
sharedcount	19
sitemapxml_check	20
urlInSitemap	20
url_with_http	21
w3cValidate	21

Index **22**

allowedByRobots	<i>Function to check if URL is blocked by robots.txt</i>
-----------------	--

Description

This function checks if a given URL is blocked by the robots.txt file of the website.

Usage

```
allowedByRobots(url, bot = "googlebot")
```

Arguments

url	The url you want to check
bot	The bot you want to check the indexability with. Default is googlebot allowed-ByRobots()

Examples

```
allowedByRobots("https://www.r-project.org/", bot = "googlebot")
```

choice	<i>Check general choices</i>
--------	------------------------------

Description

Check general choices

Usage

```
choice(choice, possible = "")
```

Arguments

choice	The choice you want to analyze
possible	What are good inputs? choice()

domainAge	<i>Function to retrieve the Age of a given Domain</i>
-----------	---

Description

This function allows you to get the Domain Age for a given Domain. (Attention:At the moment just possible with .com-Domains)

Usage

```
domainAge(domain)
```

Arguments

domain	The domain you want to get the age for. domainAge()
--------	---

Examples

```
## Not run:
domainAge("https://amazon.com")
```

```
## End(Not run)
```

downloadSitemap	<i>Function to download the Sitemap.xml of a given Webpage</i>
-----------------	--

Description

This function allows to Download the Sitemap.xml of a given Webpage

Usage

```
downloadSitemap(sitemap)
```

Arguments

sitemap	The Sitemap you want to download. This can also be a Index-Sitemap downloadSitemap()
---------	--

Examples

```
downloadSitemap("http://ohren-reinigen.de/sitemap.xml")
```

extractLinks	<i>Function to extract all links of a given Url.</i>
--------------	--

Description

This function allows you to extract all links of a given URL. You can specify if you just want unique Links or all. The output is a Data Frame.

Usage

```
extractLinks(url, linkType = "all", uniqueLinks = FALSE)
```

Arguments

url	The url you want to extract the Links of
linkType	The LinkType you want to analyze. All Links, Just Internal Links or Links to other Domains (external). You can specify "all","external" and "internal". Default is "all"
uniqueLinks	Logical. Extract unique Links as one is TRUE. Default is FALSE extractLinks()

Examples

```
extractLinks("https://www.r-project.org/", uniqueLinks = FALSE)
```

getBingResults	<i>Function to retrieve data from Google Suggest for a specific keyword</i>
----------------	---

Description

The function allows you to get the search results from Bing. The Function extracts just the organic Search results. the Output is a Dataframe containing, Title,Description, URL and Landing Page.

Usage

```
getBingResults(keyword, results = 10)
```

Arguments

keyword	The initial keyword you want the results for
results	Number of results you want to get. 10 is the Limit. getBingResults()

Examples

```
getBingResults("R Project")
```

googleSuggest	<i>Function to retrieve data from Google Suggest for a specific keyword</i>
---------------	---

Description

This function allows you to get more longtail Keywords for a specific given Keyword. The Tool uses the Google Autocomplete function to retrieve this keywords.

Usage

```
googleSuggest(keyword, language = "en", walkThrough = FALSE,
  questions = FALSE, prepositions = FALSE, comparisons = FALSE)
```

Arguments

keyword	The initial keyword you want to get suggestions for
language	The language you want to get suggestions. Default is en.
walkThrough	logical Input. If TRUE the function adds every letter before and after the keyword to loop through the Google Autocomplete.
questions	Logical. Checks for keywords with question-Phrases like "How much is beer"
prepositions	Logical. Checks for keywords with prepositions-Phrases like "beer without alcohol"
comparisons	Logical. Checks keywords with comparison phrase like "beer versus wine" googleSuggest()

Examples

```
googleSuggest("R Project", language = "en", walkThrough = FALSE)
```

host_tld	<i>Check if the User just provided the Hostname and TLD</i>
----------	---

Description

Check if the User just provided the Hostname and TLD

Usage

```
host_tld(url)
```

Arguments

url	The URL you want to analyze
-----	-----------------------------

hrefLang	<i>Function to retrieve the hrefLang-Tag for a given URL</i>
----------	--

Description

This function allows you to get the hrefLang-Tags for a given URL. You get the Output as a Data Frame containing the Link and the Country/Language specification.

Usage

```
hrefLang(url)
```

Arguments

url	The URL you want to get the hrefLang-Tag for hrefLang()
-----	---

Examples

```
hrefLang("https://www.r-project.org/")
```

htag	<i>Function to retrieve the specified H-Tag of a URL as a Data Frame</i>
------	--

Description

This function allows you to get the H-Tags you specified in the Call of a given URL. The function returns the H-Tags as Data Frame.

Usage

```
htag(url, hTag = "h1")
```

Arguments

url	The URL you want to get the H-Tags for
hTag	The H-Tags you want to get (exampale = h2-Tags). Default is H1 htag()

Examples

```
htag("https://www.r-project.org/", hTag = "h1")
```

htag_count	<i>Function to retrieve the number of specified H-Tag for a given URL</i>
------------	---

Description

This function allows you to get the number of H-Tags you specified in the Call for a given URL.

Usage

```
htag_count(url, hTag = "h1")
```

Arguments

url	The URL you want to get the H-Tags for
hTag	The H-Tags you want to get (exampale = h2-Tags). Default is H1 htag_count()

Examples

```
htag_count("https://www.r-project.org/", hTag = "h3")
```

htag_input *Check h-Tag Input*

Description

Check h-Tag Input

Usage

```
htag_input(h_tag)
```

Arguments

h_tag The h_tag you want to analyze

HTMLcanonical *Function to retrieve the Canonical Link element for a given URL*

Description

This function allows you to get the canonical Link element for a given URL.

Usage

```
HTMLcanonical(url)
```

Arguments

url The URL you want to get the Canonical-Tag for HTMLcanonical()

Examples

```
## Not run:  
HTMLcanonical("https://www.r-project.org/")  
  
## End(Not run)
```

HTMLdescription	<i>Function to retrieve the Meta-Description of a given URL</i>
-----------------	---

Description

This function allows you to get the Meta-Description of a specific URL.

Usage

```
HTMLdescription(url)
```

Arguments

`url` The URL you want to get the description for HTMLdescription()

Examples

```
HTMLdescription("https://www.r-project.org/")
```

HTMLdescription_length	<i>Function to retrieve the number of characters a Meta Description contains for a given URL</i>
------------------------	--

Description

This function allows you to get the number of characters in a Meta Description for a given URL.

Usage

```
HTMLdescription_length(url)
```

Arguments

`url` The URL you want to get the description length for HTMLdescription_length()

Examples

```
HTMLdescription_length("https://www.r-project.org/")
```

HTMLrobots

Function to retrieve the Meta Robots for a given URL

Description

This function allows you to get the Meta Robots for a given URL.

Usage

```
HTMLrobots(url)
```

Arguments

url The URL you want to get the Meta Robots for HTMLrobots()

Examples

```
HTMLrobots("https://www.r-project.org/")
```

HTMLtitle

Function to retrieve the Title of a given URL

Description

This function allows you to get the title of a specific URL.

Usage

```
HTMLtitle(url)
```

Arguments

url The URL you want to get the title for HTMLtitle()

Examples

```
HTMLtitle("https://www.r-project.org/")
```

HTMLtitle_length	<i>Function to retrieve the number of characters a Title-Tag contains for a given URL</i>
------------------	---

Description

This function allows you to get the number of characters in a Title-Tag for a given URL.

Usage

```
HTMLtitle_length(url)
```

Arguments

url	The URL you want to get the title length for HTMLtitle_length()
-----	---

Examples

```
HTMLtitle_length("https://www.r-project.org/")
```

internal_all_file	<i>Check Screaming-Frog Input Files</i>
-------------------	---

Description

Check Screaming-Frog Input Files

Usage

```
internal_all_file(input)
```

Arguments

input	The input you want to analyze
-------	-------------------------------

isIndexable *Function to check if a URL is indexable*

Description

This function checks if a given URL is indexable. Therefore it checks different rules: Status-Code: Does the URI gives a correct Status-Code (200) Canonical-Link: Is the canonical Link Element implementet, so the URL can be indexed? Meta-Robots: Is the Meta-Robots blocking the ULR? robots.txt: Is the robots.txt blocking the URL from crawling?

Usage

```
isIndexable(url, bot = "googlebot", errorType = TRUE)
```

Arguments

url	The url you want to check the indexability for
bot	The bot you want to check the indexability with. Default is googlebot
errorType	Logical. Retrieves the Error, why the URL can not be indexed. Attention not all rules at once are displayed. isIndexable()

Examples

```
isIndexable("https://www.r-project.org/", bot = "googlebot", errorType = TRUE)
```

keywordResults *Function to retrieve the Number of results for a given Keyword*

Description

This function allows you to get the number of Results for a given Keyword. You can get the value even from Google or Bing, by specifying this in the Call.

Usage

```
keywordResults(keyword, searchengine = "google")
```

Arguments

keyword	The keyword you want to get the number of results for
searchengine	The Searchengine you want to get the results form. Even Google or Bing. Default is Google. keywordResults()

Examples

```
keywordResults("R Project", searchengine = "google")
```

lastCached	<i>Function to get the last Google Cache Date for a URL</i>
------------	---

Description

This function allows you to get the latest Google Cache Date for a given URL.

Usage

```
lastCached(url)
```

Arguments

url	The url you want to get the Cache Date lastCached()
-----	---

Examples

```
lastCached("https://www.r-project.org/")
```

linkCount	<i>Function to retrieve the number of outgoing links for a given Url.</i>
-----------	---

Description

This function allows you to get the number of outgoing links for a given URL. You can specify if you want to count same Links as one or not.

Usage

```
linkCount(url, linkType = "all", uniqueLinks = FALSE)
```

Arguments

url	The url you want to get the number of outgoing links for
linkType	The LinkType you want to analyze. All Links, Just Internal Links or Links to other Domains (external). You can specify "all", "external" and "internal". Default is "all"
uniqueLinks	Logical. Count unique Links as one is TRUE. Default is FALSE linkCount()

Examples

```
linkCount("https://www.r-project.org/", linkType = "all", uniqueLinks = FALSE)
```

mobileFriendly *Function to retrieve Data from Googles Mobile Friendly Testing API*

Description

This function allows you to retrieve Data from Googles Mobile Friendly Test API. The Function retrieves the data for a given Url. The API provides you a TRUE for "is mobile friendly" and FALSE for "is not mobile friendly"

Usage

```
mobileFriendly(domain, APIToken = "AIzaSyC1gUZEsqk-nny6f3KnJyTKuy3006fpGCw")
```

Arguments

domain	The Url you want to perform the test for.
APIToken	Your API Token for this Google Project. Please use your own token if you perform a lot of calls. mobileFriendly()

Examples

```
## Not run:
mobileFriendly("https://www.r-project.org/")

## End(Not run)
```

mozLinkMetrics *Function to retrieve Link Metrics from Moz-API*

Description

This function allows you to get the Moz Link Metrics. You need a API Key to retrieve this Data. You can generate one for free here: <https://moz.com/help/guides/moz-api/mozscape/getting-started-with-mozscape/create-and-manage-your-account> At the Moment you just get the Data provided by a free Account.

Usage

```
mozLinkMetrics(page, Access_ID, Secret_Key, Scope = "page_to_page",
  Limit = 1, Filter = "", Sort = "", SourceCols = "536870916",
  TargetCols = "536870916", LinkCols = "")
```

Arguments

page	The page you want the data for
Access_ID	Your Moz Access ID
Secret_Key	Your Moz Secret Key
Scope	indicates which links to return based on whether the target and source URLs are pages, domains, or subdomains.
Limit	The default value of Limit is 1 and the maximum value is 50.
Filter	excludes results unless they meet criteria you specify. You can specify more than one value by separating multiple Filter parameters with a plus symbol +.
Sort	Sort links results by: https://moz.com/help/guides/moz-api/mozscape/api-reference/link-metrics
SourceCols	is a bit flag: a numerical value specifying which data to include about source URLs that link to the target URL.
TargetCols	is a bit flag: a numerical value specifying which data to include about the target URL.
LinkCols	is a bit flag: a numerical value specifying which data to include about the link itself (for example, if the link is nofollowed). mozLinkMetrics()

Examples

```
## Not run:
page<-"https://www.r-project.org/"
Access_ID<-{{YOUR_ACCESS_ID}}
Secret_Key<-{{YOUR_SECRET_KEY}}
mozLinkMetrics(page, Access_ID, Secret_Key)

## End(Not run)
```

 mozUrlMetrics

Function to retrieve URL Metrics from Moz-API

Description

This function allows you to get the Moz URL Metrics. You need a API Key to retrieve this Data. You can generate one for free here: <https://moz.com/help/guides/moz-api/mozscape/getting-started-with-mozscape/create-and-manage-your-account> At the Moment you just get the Data provided by a free Account.

Usage

```
mozUrlMetrics(page, Access_ID, Secret_Key)
```

Arguments

page	The page you want the data for
Access_ID	Your Moz Access ID
Secret_Key	Your Moz Secret Key mozUrlMetrics()

Examples

```
## Not run:
page<-"https://www.r-project.org/"
Access_ID<-{{YOUR_ACCESS_ID}}
Secret_Key<-{{YOUR_SECRET_KEY}}
mozUrlMetrics(page, Access_ID, Secret_Key)

## End(Not run)
```

pagesInIndex	<i>Function to retrieve the number of indexed URLs for a given Domain or URL-String</i>
--------------	---

Description

This function allows you to get the number of indexed URLs for a given Domain or URL-String. The Function performs a Google-Site request to gets the value.

Usage

```
pagesInIndex(url)
```

Arguments

url	The url you want to get the number of results for pagesInIndex()
-----	--

Examples

```
pagesInIndex("https://www.r-project.org/")
```

pageSpeed *Function to retrieve Data from Googles Pagespeed Testing API*

Description

This function allows you to retrieve Data from Googles Pagespeed Testing API. The Function retrieves the data (a numeric value between 0 and 100) for a given Url.

Usage

```
pageSpeed(url, APIToken = "AIzaSyC1gUZEsqk-nny6f3KnJyTKuy3006fpGCw",
  strategy = "mobile", pageStats = FALSE, speedScore = FALSE)
```

Arguments

url	The Url you want to perform the test for.
APIToken	Your API Token for this Google Project. Please use your own token if you perform a lot of calls.
strategy	Character. Choose if you want to get the data for "mobile" or "desktop"
pageStats	Logical. Do you want to get the Pagestats from the API? Default is TRUE
speedScore	Logical. If TRUE you just get the Pagespeed Score Number. Default is FALSE

pageSpeed()

Examples

```
pageSpeed("https://www.r-project.org/")
```

responseCode *Function to get the Status Code of a given URL*

Description

This function allows you to get the the status Code of a given URL

Usage

```
responseCode(url)
```

Arguments

url	The URL you want to get the status Code for responseCode()
-----	--

Examples

```
responseCode("https://www.r-project.org/")
```

screamingfrog_crawlVsSitemap

Function to compare the crawled URLs with the URLs in the sitemap.xml

Description

This function allows to compare the crawled URLs with the URLs in the sitemap.xml. The Output are the missing URLs - in the Sitemap or in the Crawl

Usage

```
screamingfrog_crawlVsSitemap(crawl, sitemap, deltaIn = "sitemap",
    checkImages = FALSE)
```

Arguments

crawl	The Path to the exported Screaming Frog csv-File
sitemap	The Sitemap you want to compare with the Crawl
deltaIn	The Delta you want to analyze. Do you want to get the URLs in the Crawl missing in the Sitemap = "crawl" or the URLs in the Sitemap.xml not found in the Crawl = "sitemap"
checkImages	Logical. Do you want to check the Images as well. Default is FALSE. screamingfrog_crawlVsSitemap()

screamingfrog_internalPagerank

Function to calculate the internal PageRank of a crawled Page

Description

This function allows you to calculate the internal PageRank of a given Domain. The calculation is based on a ScreamingFrog Crawl and the export "all_outlinks.csv".

Usage

```
screamingfrog_internalPagerank(crawl, domain = NULL)
```

Arguments

crawl	The Path to your "all_outlinks.csv"-File
domain	The function also analyzes outgoing links. If you want the results just for your domain. Specify it here. screamingfrog_internalPagerank()

seoDiver	<i>Function to retrieve data from the SEO Diver API</i>
----------	---

Description

This function allows you to get Data from the SEO Diver API. <http://de.seodiver.com/api> Note: This Data is only available for DE,CH,AT Search Results. You get "Suchreichweite", "Suchwahrnehmung" and "Statische Sichtbarkeit"

Usage

```
seoDiver(url, type = "Suchreichweite")
```

Arguments

url	The URL you want to get the Data for
type	The type of Data you want to get. Possible Inputs are: "Suchreichweite", "Suchwahrnehmung" and "Statische Sichtbarkeit" seoDiver()

Examples

```
seoDiver("r-project.org", type = "Suchreichweite")
```

sharedcount	<i>Function to get the Social Shares for a given URL</i>
-------------	--

Description

This function allows to get the shares of a given Domain into R. Therefore the function uses the Sharedcount-API. To use the function you have to set up a free API-Account here: <https://www.sharedcount.com/>

Usage

```
sharedcount(url, key)
```

Arguments

url	The URL you want to analyze
key	Your API-Key you get from https://www.sharedcount.com/ fro free sharedcount()

Examples

```
## Not run:
url<-"https://www.r-project.org/"
key<-{{YOUR_API_KEY}}
sharedcount(url, key)

## End(Not run)
```

sitemapxml_check *Check if the User provided a correct Sitemap.xml*

Description

Check if the User provided a correct Sitemap.xml

Usage

```
sitemapxml_check(sitemap)
```

Arguments

sitemap The sitemap you want to analyze

urlInSitemap *Function to check if a given URL is in a XML-Sitemap*

Description

This function allows you to check if a given URL is found in a XML-Sitemap. For this you can also put in your Index-Sitemap.

Usage

```
urlInSitemap(url, sitemap)
```

Arguments

url The URL you want to check
sitemap The Link to the Sitemap you want to find the URL in urlInSitemap()

Examples

```
urlInSitemap("http://ohren-reinigen.de/", "http://ohren-reinigen.de/sitemap.xml")
```

url_with_http	<i>URL Input checks</i>
---------------	-------------------------

Description

URL Input checks

Usage

```
url_with_http(url)
```

Arguments

url	The URL you want to analyze
-----	-----------------------------

w3cValidate	<i>Function to perform a W3C Validation Test for a specific URL</i>
-------------	---

Description

This function allows you to perform a W3C Validation test for a given URL. You will get the number of Errors and Warnings for the URL as Data Frame.

Usage

```
w3cValidate(url)
```

Arguments

url	The url you want to perform the Validation for w3cValidate()
-----	--

Examples

```
w3cValidate("https://www.r-project.org/")
```

Index

- *Topic **host_tld()**
 - host_tld, 6
- *Topic **htag_input()**
 - htag_input, 8
- *Topic **internal.**
 - host_tld, 6
 - htag_input, 8
 - internal_all_file, 11
 - sitemapxml_check, 20
 - url_with_http, 21
- *Topic **internal_all_file()**
 - internal_all_file, 11
- *Topic **sitemapxml_check()**
 - sitemapxml_check, 20
- *Topic **url_with_http()**
 - url_with_http, 21
- allowedByRobots, 2
- choice, 3
- domainAge, 3
- downloadSitemap, 4
- extractLinks, 4
- getBingResults, 5
- googleSuggest, 5
- host_tld, 6
- hrefLang, 6
- htag, 7
- htag_count, 7
- htag_input, 8
- HTMLcanonical, 8
- HTMLdescription, 9
- HTMLdescription_length, 9
- HTMLrobots, 10
- HTMLtitle, 10
- HTMLtitle_length, 11
- internal_all_file, 11
- isIndexable, 12
- keywordResults, 12
- lastCached, 13
- linkCount, 13
- mobileFriendly, 14
- mozLinkMetrics, 14
- mozUrlMetrics, 15
- pagesInIndex, 16
- pageSpeed, 17
- responseCode, 17
- screamingfrog_crawlVsSitemap, 18
- screamingfrog_internalPagerank, 18
- seoDiver, 19
- sharedcount, 19
- sitemapxml_check, 20
- url_with_http, 21
- urlInSitemap, 20
- w3cValidate, 21