Package 'robotstxt'

July 18, 2018

Date 2018-07-18

Type Package

Title A 'robots.txt' Parser and 'Webbot'/'Spider'/'Crawler' Permissions Checker

Version 0.6.2

Description Provides functions to download and parse 'robots.txt' files. Ultimately the package makes it easy to check if bots (spiders, crawler, scrapers, ...) are allowed to access specific resources on a domain.

License MIT + file LICENSE

LazyData TRUE

BugReports https://github.com/ropensci/robotstxt/issues

URL https://github.com/ropensci/robotstxt

Imports stringr (>= 1.0.0), httr (>= 1.0.0), spiderbar (>= 0.2.0), future (>= 1.6.2), future.apply (>= 1.0.0), magrittr, utils

Suggests knitr, rmarkdown, dplyr, testthat, covr

Depends R (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Peter Meissner [aut, cre], Oliver Keys [ctb], Rich Fitz John [ctb]

Maintainer Peter Meissner <retep.meissner@gmail.com>

Repository CRAN

Date/Publication 2018-07-18 21:30:03 UTC

R topics documented:

`get_robotstxt` *downloading robots.txt file*

Description

downloading robots.txt file

Usage

```
get_robotstxt(domain, warn = TRUE, force = FALSE,
  user_agent = utils::sessionInfo()$R.version$version.string,
  ssl_verifypeer = c(1, 0))
```

Arguments

`domain`	domain from which to download robots.txt file
`warn`	warn about being unable to download domain/robots.txt because of
`force`	if TRUE instead of using possible cached results the function will re-download the robotstxt file HTTP response status 404. If this happens,
`user_agent`	HTTP user-agent string to be used to retrieve robots.txt file from domain
`ssl_verifypeer`	analog to CURL option https://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYPEER.html – and might help with robots.txt file retrieval in some cases

get_robotstxts *function to get multiple robotstxt files*

Description

function to get multiple robotstxt files

Usage

```
get_robotstxts(domain, warn = TRUE, force = FALSE,
  user_agent = utils::sessionInfo()$R.version$version.string,
  ssl_verifypeer = c(1, 0), use_futures = FALSE)
```

Arguments

domain	domain from which to download robots.txt file
warn	warn about being unable to download domain/robots.txt because of
force	if TRUE instead of using possible cached results the function will re-download the robotstxt file HTTP response status 404. If this happens,
user_agent	HTTP user-agent string to be used to retrieve robots.txt file from domain
ssl_verifypeer	analog to CURL option https://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYPEER.html – and might help with robots.txt file retrieval in some cases #'
use_futures	Should future::future_lapply be used for possible parallel/async retrieval or not. Note: check out help pages and vignettes of package future on how to set up plans for future execution because the robotstxt package does not do it on its own.

get_robotstxt_http_get *get_robotstxt() worker function to execute HTTP request*

Description

get_robotstxt() worker function to execute HTTP request

Usage

```
get_robotstxt_http_get(domain,
  user_agent = utils::sessionInfo()$R.version$version.string,
  ssl_verifypeer = 1)
```

Arguments

domain	domain from which to download robots.txt file
user_agent	HTTP user-agent string to be used to retrieve robots.txt file from domain
ssl_verifypeer	analog to CURL option https://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYPEER.html – and might help with robots.txt file retrieval in some cases

---

guess_domain *function guessing domain from path*

Description

function guessing domain from path

Usage

```
guess_domain(x)
```

Arguments

x	path aka URL from which to infer domain

---

is_valid_robotstxt *function that checks if file is valid / parsable robots.txt file*

Description

function that checks if file is valid / parsable robots.txt file

Usage

```
is_valid_robotstxt(text)
```

Arguments

text	content of a robots.txt file provides as character vector

parse_robotstxt *function parsing robots.txt*

Description

function parsing robots.txt

Usage

```
parse_robotstxt(txt)
```

Arguments

txt	content of the robots.txt file

Value

a named list with useragents, comments, permissions, sitemap

paths_allowed *check if a bot has permissions to access page(s)*

Description

check if a bot has permissions to access page(s)

Usage

```
paths_allowed(paths = "/", domain = "auto", bot = "*",
  user_agent = utils::sessionInfo()$R.version$version.string,
  check_method = c("spiderbar"), warn = TRUE, force = FALSE,
  ssl_verifypeer = c(1, 0), use_futures = TRUE, robotstxt_list = NULL)
```

Arguments

paths	paths for which to check bot's permission, defaults to "/"
domain	Domain for which paths should be checked. Defaults to "auto". If set to "auto" function will try to guess the domain by parsing the paths argument. Note however, that these are educated guesses which might utterly fail. To be on the save side, provide appropriate domains manually.
bot	name of the bot, defaults to "*"
user_agent	HTTP user-agent string to be used to retrieve robots.txt file from domain
check_method	at the moment only kept for backward compatibility reasons - do not use parameter anymore –> will let the function simply use the default

warn	warn about being unable to download domain/robots.txt because of
force	if TRUE instead of using possible cached results the function will re-download the robotstxt file HTTP response status 404. If this happens,
ssl_verifypeer	analog to CURL option https://curl.haxx.se/libcurl/c/CURLOPT_SSL_VERIFYPEER.html – and might help with robots.txt file retrieval in some cases
use_futures	Should future::future_lapply be used for possible parallel/async retrieval or not. Note: check out help pages and vignettes of package future on how to set up plans for future execution because the robotstxt package does not do it on its own.
robotstxt_list	either NULL – the default – or a list of character vectors with one vector per path to check

---

`paths_allowed_worker_spiderbar` *paths_allowed_worker spiderbar flavor*

Description

paths_allowed_worker spiderbar flavor

Usage

```
paths_allowed_worker_spiderbar(domain, bot, paths, robotstxt_list)
```

Arguments

domain	Domain for which paths should be checked. Defaults to "auto". If set to "auto" function will try to guess the domain by parsing the paths argument. Note however, that these are educated guesses which might utterly fail. To be on the save side, provide appropriate domains manually.
bot	name of the bot, defaults to "*"
paths	paths for which to check bot's permission, defaults to "/"
robotstxt_list	either NULL – the default – or a list of character vectors with one vector per path to check

print.robotstxt *printing robotstxt*

Description

printing robotstxt

Usage

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

Arguments

x	robotstxt instance to be printed
...	goes down the sink

print.robotstxt_text *printing robotstxt_text*

Description

printing robotstxt_text

Usage

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

Arguments

x	character vector aka robotstxt$text to be printed
...	goes down the sink

remove_domain *function to remove domain from path*

Description

function to remove domain from path

Usage

```
remove_domain(x)
```

Arguments

x	path aka URL from which to first infer domain and then remove it

robotstxt *Generate a representations of a robots.txt file*

Description

The function generates a list that entails data resulting from parsing a robots.txt file as well as a function called check that enables to ask the representation if bot (or particular bots) are allowed to access a resource on the domain.

Usage

```
robotstxt(domain = NULL, text = NULL, user_agent = NULL, warn = TRUE,
  force = FALSE)
```

Arguments

domain	Domain for which to generate a representation. If text equals to NULL, the function will download the file from server - the default.
text	If automatic download of the robots.txt is not preferred, the text can be supplied directly.
user_agent	HTTP user-agent string to be used to retrieve robots.txt file from domain
warn	warn about being unable to download domain/robots.txt because of
force	if TRUE instead of using possible cached results the function will re-download the robotstxt file HTTP response status 404. If this happens,

Value

Object (list) of class robotstxt with parsed data from a robots.txt (domain, text, bots, permissions, host, sitemap, other) and one function to (check()) to check resource permissions.

Fields

`domain` character vector holding domain name for which the robots.txt file is valid; will be set to NA if not supplied on initialization

`text` character vector of text of robots.txt file; either supplied on initialization or automatically downloaded from domain supplied on initialization

`bots` character vector of bot names mentioned in robots.txt

`permissions` data.frame of bot permissions found in robots.txt file

`host` data.frame of host fields found in robots.txt file

`sitemap` data.frame of sitemap fields found in robots.txt file

`other` data.frame of other - none of the above - fields found in robots.txt file

`check()` Method to check for bot permissions. Defaults to the domains root and no bot in particular. check() has two arguments: paths and bot. The first is for supplying the paths for which to check permissions and the latter to put in the name of the bot.

Examples

```
## Not run:
rt <- robotstxt(domain="google.com")
rt$bots
rt$permissions
rt$check( paths = c("/", "forbidden"), bot="*")

## End(Not run)
```

---

rt_cache *get_robotstxt() cache*

Description

get_robotstxt() cache

Usage

```
rt_cache
```

Format

An object of class `environment` of length 0.

%>% *re-export magrittr pipe operator*

Description

re-export magrittr pipe operator

Index