Package: robotstxt (via r-universe)

October 29, 2026

Type Package

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

Version 0.7.15.9000

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

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

URL https://docs.ropensci.org/robotstxt/, https://github.com/ropensci/robotstxt

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

Suggests knitr, rmarkdown, dplyr, testthat (>= 3.0.0), covr, curl

Depends R (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.2

Encoding UTF-8

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2024-11-15 19:32:37 UTC

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

RemoteRef HEAD

RemoteSha d3d0a4d525e87204010d10162b01a921b05a6149

Contents

%>% . . . 2
as.list.robotstxt_text . . . 2
fix_url . . . 3
get_robotstxt . . . 3
get_robotstxts . . . 4
guess_domain . . . 6
http_domain_changed . . . 6
http_subdomain_changed . . . 7
http_was_redirected . . . 7
is_suspect_robotstxt . . . 8
is_valid_robotstxt . . . 8
list_merge . . . 9
null_to_default . . . 9
parse_robotstxt . . . 10
paths_allowed . . . 10
paths_allowed_worker_spiderbar . . . 12
print.robotstxt . . . 12
print.robotstxt_text . . . 13
remove_domain . . . 13
request_handler_handler . . . 14
robotstxt . . . 14
rt_cache . . . 16
rt_last_http . . . 17
rt_request_handler . . . 17

Index **20**

`%>%` *re-export magrittr pipe operator*

Description

re-export magrittr pipe operator

`as.list.robotstxt_text` *Convert robotstxt_text to list*

Description

Convert robotstxt_text to list

Usage

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

Arguments

x	class robotstxt_text object to be transformed into list
...	further arguments (inherited from base::as.list())

---

fix_url *Add http protocal if missing from URL*

Description

Add http protocal if missing from URL

Usage

```
fix_url(url)
```

Arguments

url	a character string containing a single URL

---

get_robotstxt *Download a robots.txt file*

Description

Download a robots.txt file

Usage

```
get_robotstxt(
  domain,
  warn = getOption("robotstxt_warn", TRUE),
  force = FALSE,
  user_agent = utils::sessionInfo()$R.version$version.string,
  ssl_verifypeer = c(1, 0),
  encoding = "UTF-8",
  verbose = FALSE,
  rt_request_handler = robotstxt::rt_request_handler,
  rt_robotstxt_http_getter = robotstxt::get_robotstxt_http_get,
  on_server_error = on_server_error_default,
  on_client_error = on_client_error_default,
```

```
  on_not_found = on_not_found_default,
  on_redirect = on_redirect_default,
  on_domain_change = on_domain_change_default,
  on_file_type_mismatch = on_file_type_mismatch_default,
  on_suspect_content = on_suspect_content_default
)
```

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	either 1 (default) or 0, if 0 it disables SSL peer verification, which might help with robots.txt file retrieval
encoding	Encoding of the robots.txt file.
verbose	make function print out more information
rt_request_handler	handler function that handles request according to the event handlers specified
rt_robotstxt_http_getter	function that executes HTTP request
on_server_error	request state handler for any 5xx status
on_client_error	request state handler for any 4xx HTTP status that is not 404
on_not_found	request state handler for HTTP status 404
on_redirect	request state handler for any 3xx HTTP status
on_domain_change	request state handler for any 3xx HTTP status where domain did change as well
on_file_type_mismatch	request state handler for content type other than 'text/plain'
on_suspect_content	request state handler for content that seems to be something else than a robots.txt file (usually a JSON, XML or HTML)

---

get_robotstxts *Download multiple robotstxt files*

Description

Download 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,
  verbose = FALSE,
  rt_request_handler = robotstxt::rt_request_handler,
  rt_robotstxt_http_getter = robotstxt::get_robotstxt_http_get,
  on_server_error = on_server_error_default,
  on_client_error = on_client_error_default,
  on_not_found = on_not_found_default,
  on_redirect = on_redirect_default,
  on_domain_change = on_domain_change_default,
  on_file_type_mismatch = on_file_type_mismatch_default,
  on_suspect_content = on_suspect_content_default
)
```

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	either 1 (default) or 0, if 0 it disables SSL peer verification, which might help with robots.txt file retrieval
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.
verbose	make function print out more information
rt_request_handler	handler function that handles request according to the event handlers specified
rt_robotstxt_http_getter	function that executes HTTP request
on_server_error	request state handler for any 5xx status
on_client_error	request state handler for any 4xx HTTP status that is not 404
on_not_found	request state handler for HTTP status 404
on_redirect	request state handler for any 3xx HTTP status

on_domain_change
: request state handler for any 3xx HTTP status where domain did change as well

on_file_type_mismatch
: request state handler for content type other than 'text/plain'

on_suspect_content
: request state handler for content that seems to be something else than a robots.txt file (usually a JSON, XML or HTML)

---

guess_domain *Guess a domain from path*

Description

Guess a domain from path

Usage

```
guess_domain(x)
```

Arguments

x	path aka URL from which to infer domain

---

http_domain_changed *Check if HTTP domain changed*

Description

Check if HTTP domain changed

Usage

```
http_domain_changed(response)
```

Arguments

response	an httr response object, e.g. from a call to httr::GET()

Value

logical of length 1 indicating whether or not any domain change happened during the HTTP request

http_subdomain_changed *Check if HTTP subdomain changed*

Description

Check if HTTP subdomain changed

Usage

```
http_subdomain_changed(response)
```

Arguments

response	an httr response object, e.g. from a call to httr::GET()

Value

logical of length 1 indicating whether or not any subdomain change happened during the HTTP request

http_was_redirected *Check if HTTP redirect occurred*

Description

Check if HTTP redirect occurred

Usage

```
http_was_redirected(response)
```

Arguments

response	an httr response object, e.g. from a call to httr::GET()

Value

logical of length 1 indicating whether or not any redirect happened during the HTTP request

is_suspect_robotstxt *Check if file is valid / parsable robots.txt file*

Description

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

Usage

```
is_suspect_robotstxt(text)
```

Arguments

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

is_valid_robotstxt *Validate if a file is valid / parsable robots.txt file*

Description

Validate if a file is valid / parsable robots.txt file

Usage

```
is_valid_robotstxt(text, check_strickt_ascii = FALSE)
```

Arguments

text	content of a robots.txt file provided as character vector
check_strickt_ascii	whether or not to check if content does adhere to the specification of RFC to use plain text aka ASCII

list_merge *Merge a number of named lists in sequential order*

Description

Merge a number of named lists in sequential order

Usage

```
list_merge(...)
```

Arguments

...	named lists

Details

List merging is usually useful in the merging of program settings or configuraion with multiple versions across time, or multiple administrative levels. For example, a program settings may have an initial version in which most keys are defined and specified. In later versions, partial modifications are recorded. In this case, list merging can be useful to merge all versions of settings in release order of these versions. The result is an fully updated settings with all later modifications applied.

Author(s)

Kun Ren <mail@renkun.me>

The function merges a number of lists in sequential order by `modifyList`, that is, the later list always modifies the former list and form a merged list, and the resulted list is again being merged with the next list. The process is repeated until all lists in `...` or `list` are exausted.

null_to_default *Return default value if NULL*

Description

Return default value if NULL

Usage

```
null_to_default(x, d)
```

Arguments

x	value to check and return
d	value to return in case x is NULL

parse_robotstxt *Parse a robots.txt file*

Description

Parse a robots.txt file

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 = getOption("robotstxt_warn", TRUE),
  force = FALSE,
  ssl_verifypeer = c(1, 0),
  use_futures = TRUE,
  robotstxt_list = NULL,
  verbose = FALSE,
  rt_request_handler = robotstxt::rt_request_handler,
  rt_robotstxt_http_getter = robotstxt::get_robotstxt_http_get,
  on_server_error = on_server_error_default,
  on_client_error = on_client_error_default,
  on_not_found = on_not_found_default,
  on_redirect = on_redirect_default,
```

```
  on_domain_change = on_domain_change_default,
  on_file_type_mismatch = on_file_type_mismatch_default,
  on_suspect_content = on_suspect_content_default
)
```

Arguments

paths	paths for which to check bot's permission, defaults to "/". Please note that path to a folder should end with a trailing slash ("/").
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 safe 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	suppress warnings
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	either 1 (default) or 0, if 0 it disables SSL peer verification, which might help with robots.txt file retrieval
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
verbose	make function print out more information
rt_request_handler	handler function that handles request according to the event handlers specified
rt_robotstxt_http_getter	function that executes HTTP request
on_server_error	request state handler for any 5xx status
on_client_error	request state handler for any 4xx HTTP status that is not 404
on_not_found	request state handler for HTTP status 404
on_redirect	request state handler for any 3xx HTTP status
on_domain_change	request state handler for any 3xx HTTP status where domain did change as well
on_file_type_mismatch	request state handler for content type other than 'text/plain'
on_suspect_content	request state handler for content that seems to be something else than a robots.txt file (usually a JSON, XML or HTML)

paths_allowed_worker_spiderbar *Check if a spiderbar bot has permissions to access page(s)*

Description

Check if a spiderbar bot has permissions to access page(s)

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 safe side, provide appropriate domains manually.
bot	name of the bot, defaults to "*"
paths	paths for which to check bot's permission, defaults to "/". Please note that path to a folder should end with a trailing slash ("/").
robotstxt_list	either NULL – the default – or a list of character vectors with one vector per path to check

print.robotstxt *Print robotstxt*

Description

Print robotstxt

Usage

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

Arguments

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

print.robotstxt_text *Print robotstxt's text*

Description

Print robotstxt's 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 *Remove domain from path*

Description

Remove domain from path

Usage

```
remove_domain(x)
```

Arguments

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

request_handler_handler *Handle robotstxt handlers*

Description

Helper function to handle robotstxt handlers.

Usage

```
request_handler_handler(request, handler, res, info = TRUE, warn = TRUE)
```

Arguments

request	the request object returned by call to httr::GET()
handler	the handler either a character string entailing various options or a function producing a specific list, see return.
res	a list with elements '[handler names], ...', 'rtxt', and 'cache'
info	info to add to problems list
warn	if FALSE warnings and messages are suppressed

Value

a list with elements '[handler name]', 'rtxt', and 'cache'

robotstxt *Generate a representation 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 = getOption("robotstxt_warn", TRUE),
  force = FALSE,
  ssl_verifypeer = c(1, 0),
  encoding = "UTF-8",
```

```
  verbose = FALSE,
  on_server_error = on_server_error_default,
  on_client_error = on_client_error_default,
  on_not_found = on_not_found_default,
  on_redirect = on_redirect_default,
  on_domain_change = on_domain_change_default,
  on_file_type_mismatch = on_file_type_mismatch_default,
  on_suspect_content = on_suspect_content_default
)
```

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,
ssl_verifypeer	either 1 (default) or 0, if 0 it disables SSL peer verification, which might help with robots.txt file retrieval
encoding	Encoding of the robots.txt file.
verbose	make function print out more information
on_server_error	request state handler for any 5xx status
on_client_error	request state handler for any 4xx HTTP status that is not 404
on_not_found	request state handler for HTTP status 404
on_redirect	request state handler for any 3xx HTTP status
on_domain_change	request state handler for any 3xx HTTP status where domain did change as well
on_file_type_mismatch	request state handler for content type other than 'text/plain'
on_suspect_content	request state handler for content that seems to be something else than a robots.txt file (usually a JSON, XML or HTML)

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

`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. Please, note that path to a folder should end with a trailing slash ("/").

Examples

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

## End(Not run)
```

---

rt_cache *Get the robotstxt cache*

Description

Get the robotstxt cache

Usage

```
rt_cache
```

Format

An object of class `environment` of length 0.

rt_last_http *Storage for HTTP request response objects*

Description

Storage for HTTP request response objects

Execute HTTP request for get_robotstxt()

Usage

```
rt_last_http

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

Arguments

domain	the domain to get robots.txt file for.
user_agent	the user agent to use for HTTP request header. Defaults to current version of R. If 'NULL' is passed, httr will use software versions for the header, such as 'libcurl/8.7.1 r-curl/5.2.3 httr/1.4.7'
ssl_verifypeer	either 1 (default) or 0, if 0 it disables SSL peer verification, which might help with robots.txt file retrieval

Format

An object of class environment of length 1.

rt_request_handler *Handle robotstxt object retrieved from HTTP request*

Description

A helper function for get_robotstxt() that will extract the robots.txt file from the HTTP request result object. It will inform get_robotstxt() if the request should be cached and which problems occurred.

Usage

```
rt_request_handler(
  request,
  on_server_error = on_server_error_default,
  on_client_error = on_client_error_default,
  on_not_found = on_not_found_default,
  on_redirect = on_redirect_default,
  on_domain_change = on_domain_change_default,
  on_sub_domain_change = on_sub_domain_change_default,
  on_file_type_mismatch = on_file_type_mismatch_default,
  on_suspect_content = on_suspect_content_default,
  warn = TRUE,
  encoding = "UTF-8"
)

on_server_error_default

on_client_error_default

on_not_found_default

on_redirect_default

on_domain_change_default

on_sub_domain_change_default

on_file_type_mismatch_default

on_suspect_content_default
```

Arguments

request	result of an HTTP request (e.g. httr::GET())
on_server_error	request state handler for any 5xx status
on_client_error	request state handler for any 4xx HTTP status that is not 404
on_not_found	request state handler for HTTP status 404
on_redirect	request state handler for any 3xx HTTP status
on_domain_change	request state handler for any 3xx HTTP status where domain did change as well
on_sub_domain_change	request state handler for any 3xx HTTP status where domain did change but only to www-sub_domain
on_file_type_mismatch	request state handler for content type other than 'text/plain'

`on_suspect_content`
: request state handler for content that seems to be something else than a robots.txt file (usually a JSON, XML or HTML)

`warn`
: suppress warnings

`encoding`
: The text encoding to assume if no encoding is provided in the headers of the response

Format

An object of class `list` of length 4.

An object of class `list` of length 4.

An object of class `list` of length 4.

An object of class `list` of length 2.

An object of class `list` of length 3.

An object of class `list` of length 2.

An object of class `list` of length 4.

An object of class `list` of length 4.

Value

a list with three items following the following schema:
`list( rtxt = "", problems = list( "redirect" = list( status_code = 301 ), "domain" = list(from_url = "...", to_url = "...") ) )`

Index

∗ **datasets**
    rt_cache, 16
    rt_last_http, 17
    rt_request_handler, 17
%>%, 2

as.list.robotstxt_text, 2

fix_url, 3

get_robotstxt, 3
get_robotstxt_http_get *(*rt_last_http*)*, 17
get_robotstxts, 4
guess_domain, 6

http_domain_changed, 6
http_subdomain_changed, 7
http_was_redirected, 7

is_suspect_robotstxt, 8
is_valid_robotstxt, 8

list_merge, 9

null_to_default, 9

on_client_error_default *(*rt_request_handler*)*, 17
on_domain_change_default *(*rt_request_handler*)*, 17
on_file_type_mismatch_default *(*rt_request_handler*)*, 17
on_not_found_default *(*rt_request_handler*)*, 17
on_redirect_default *(*rt_request_handler*)*, 17
on_server_error_default *(*rt_request_handler*)*, 17
on_sub_domain_change_default *(*rt_request_handler*)*, 17
on_suspect_content_default *(*rt_request_handler*)*, 17

parse_robotstxt, 10
paths_allowed, 10
paths_allowed_worker_spiderbar, 12
print.robotstxt, 12
print.robotstxt_text, 13

remove_domain, 13
request_handler_handler, 14
robotstxt, 14
rt_cache, 16
rt_last_http, 17
rt_request_handler, 17