Package 'spiderbar'

August 19, 2019

Type Package

Title Parse and Test Robots Exclusion Protocol Files and Rules

Version 0.2.2

Date 2019-08-18

Author Bob Rudis (bob@rud.is) [aut, cre], SEOmoz, Inc [aut]

Maintainer Bob Rudis <bob@rud.is>

Description The 'Robots Exclusion Protocol' <https://www.robotstxt.org/orig.html> documents a set of standards for allowing or excluding robot/spider crawling of different areas of site content. Tools are provided which wrap The 'rep-cpp' <https://github.com/seomoz/rep-cpp> C++ library for processing these 'robots.txt' files.

SystemRequirements C++11

NeedsCompilation yes

URL https://gitlab.com/hrbrmstr/spiderbar

BugReports https://gitlab.com/hrbrmstr/spiderbar/issues

License MIT + file LICENSE

Suggests testthat, covr, robotstxt

Depends R (>= 3.2.0)

Encoding UTF-8

Imports Rcpp

RoxygenNote 6.1.1

LinkingTo Rcpp

Repository CRAN

Date/Publication 2019-08-19 16:50:07 UTC

R topics documented:

`can_fetch` *Test URL paths against a* `robxp` `robots.txt` *object*

Description

Provide a character vector of URL paths plus optional user agent and this function will return a logical vector indicating whether you have permission to fetch the content at the respective path.

Usage

```
can_fetch(obj, path = "/", user_agent = "*")
```

Arguments

`obj`	`robxp` object
`path`	path to test
`user_agent`	user agent to test

Examples

```
gh <- paste0(readLines(system.file("extdata", "github-robots.txt",
            package="spiderbar")), collapse="\n")
gh_rt <- robxp(gh)

can_fetch(gh_rt, "/humans.txt", "*") # TRUE
can_fetch(gh_rt, "/login", "*") # FALSE
can_fetch(gh_rt, "/oembed", "CCBot") # FALSE

can_fetch(gh_rt, c("/humans.txt", "/login", "/oembed"))
```

`crawl_delays` *Retrive all agent crawl delay values in a* `robxp` `robots.txt` *object*

Description

Retrive all agent crawl delay values in a `robxp` `robots.txt` object

Usage

```
crawl_delays(obj)
```

Arguments

`obj`	`robxp` object

Value

data frame of agents and their crawl delays

Note

-1 will be returned for any listed agent *without* a crawl delay setting

Examples

```
gh <- paste0(readLines(system.file("extdata", "github-robots.txt",
            package="spiderbar")), collapse="\n")
gh_rt <- robxp(gh)
crawl_delays(gh_rt)

imdb <- paste0(readLines(system.file("extdata", "imdb-robots.txt",
              package="spiderbar")), collapse="\n")
imdb_rt <- robxp(imdb)
crawl_delays(imdb_rt)
```

---

robxp *Parse a 'robots.txt' file & create a 'robxp' object*

Description

This function takes in a single element character vector and parses it into a 'robxp' object.

Usage

```
robxp(x)
```

Arguments

x	either an atomic character vector containing a complete 'robots.txt" file _or_ a length >1 character vector that will concatenated into a single string _or_ a 'connection' object that will be passed to [readLines()], the result of which will be concatenated into a single string and parsed and the connection will be closed.

Examples

```
imdb <- paste0(readLines(system.file("extdata", "imdb-robots.txt",
              package="spiderbar")), collapse="\n")
rt <- robxp(imdb)
```

sitemaps *Retrieve a character vector of sitemaps from a parsed robots.txt object*

Description

Retrieve a character vector of sitemaps from a parsed robots.txt object

Usage

```
sitemaps(xp)
```

Arguments

xp	A robxp object

Value

charcter vector of all sitemaps found in the parsed `robots.txt` file

Examples

```
imdb <- paste0(readLines(system.file("extdata", "imdb-robots.txt",
              package="rep")), collapse="\n")
rt <- robxp(imdb)
sitemaps(rt)
```

spiderbar *Parse and Test Robots Exclusion Protocol Files and Rules*

Description

The 'Robots Exclusion Protocol' (https://www.robotstxt.org/orig.html) documents a set of standards for allowing or excluding robot/spider crawling of different areas of site content. Tools are provided which wrap The rep-cpp https://github.com/seomoz/rep-cpp C++ library for processing these 'robots.txt" files.

Author(s)

Bob Rudis (bob@rud.is)

Index