

Package ‘searcher’

February 6, 2020

Title Query Search Interfaces

Version 0.0.5

Description Provides a search interface to look up terms on 'Google', 'Bing', 'DuckDuckGo', 'Startpage', 'Twitter', 'StackOverflow', 'RStudio Community', 'GitHub', and 'BitBucket'. Upon searching, a browser window will open with the aforementioned search results.

URL <https://github.com/r-assist/searcher>

BugReports <https://github.com/r-assist/searcher/issues>

Depends R (>= 3.3.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Suggests testthat (>= 2.1.0), covr, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author James Balamuta [aut, cre] (<<https://orcid.org/0000-0003-2826-8458>>)

Maintainer James Balamuta <balamut2@illinois.edu>

Repository CRAN

Date/Publication 2020-02-06 10:40:02 UTC

R topics documented:

searcher-package	2
searcher	2
searcher-defunct	3
search_site	4

Index	8
--------------	----------

searcher-package *searcher: Query Search Interfaces*

Description

Provides a search interface to look up terms on 'Google', 'Bing', 'DuckDuckGo', 'Startpage', 'Twitter', 'StackOverflow', 'RStudio Community', 'GitHub', and 'BitBucket'. Upon searching, a browser window will open with the aforementioned search results.

Package Customizations

searcher accesses a set of default values stored in `options()` on each call to keep the function signatures small. By default, these options are given as:

- `searcher.launch_delay`: Amount of time to remain in R before opening a browser window. Default is 0.5 seconds.
- `searcher.use_rstudio_viewer`: Display search results in the RStudio viewer pane instead of a web browser. Default is FALSE.
- `searcher.default_keyword`: Suffix keyword to generate accurate results between either "base" or "tidyverse". Default is "base".
- ...

Author(s)

Maintainer: James Balamuta <balamut2@illinois.edu> ([ORCID](#))

See Also

Useful links:

- <https://github.com/r-assist/searcher>
- Report bugs at <https://github.com/r-assist/searcher/issues>

searcher *Generate a Searcher function for use with Error Handling*

Description

Constructs a function object that will search the last R error message on search portals by opening a browser.

Usage

```
searcher(site, keyword = getOption("searcher.default_keyword"))
```

Arguments

site	Name of site to search on. Supported options: "google" (default), "bing", "duckduckgo", "startpage", "rstudio community", "twitter", "stackoverflow", "github", and "bitbucket".
keyword	Opt to search under different default terms.

Details

This function acts as a closure. Thus, you will receive a function back when only specifying the site parameter. To call the function, add a second set of parentheses.

Generic Error Search

The searcher function grabs the last error message and tries to search it. This function will ensure that R language is the primary search context.

Examples

```
### Manually
searcher("google")()

## Not run:
### Automatically
# On error, automatically search the message on google
options(error = searcher("google"))

## End(Not run)
```

searcher-defunct *Defunct functions in searcher*

Description

Functions listed below are no longer included in the searcher package

Details

- [search_ixquick\(\)](#): The function binding was removed as the search engine name changed to "Startpage". Please use [search_startpage\(\)](#).

`search_site`*Search a Query on a Search Portal in a Web Browser*

Description

Creates an appropriate query string for a search engine and then opens up the resulting page in a web browser.

Usage

```
search_site(  
  query,  
  site = c("google", "bing", "duckduckgo", "ddg", "startpage", "sp",  
    "rstudio community", "rscom", "twitter", "stackoverflow", "so", "github", "gh",  
    "bitbucket", "bb"),  
  rlang = TRUE  
)
```

```
search_google(query = geterrmessage(), rlang = TRUE)
```

```
search_bing(query = geterrmessage(), rlang = TRUE)
```

```
search_duckduckgo(query = geterrmessage(), rlang = TRUE)
```

```
search_ddg(query = geterrmessage(), rlang = TRUE)
```

```
search_ixquick(query = geterrmessage(), rlang = TRUE)
```

```
search_startpage(query = geterrmessage(), rlang = TRUE)
```

```
search_sp(query = geterrmessage(), rlang = TRUE)
```

```
search_rstudio_community(query = geterrmessage(), rlang = TRUE)
```

```
search_rscom(query = geterrmessage(), rlang = TRUE)
```

```
search_twitter(query = geterrmessage(), rlang = TRUE)
```

```
search_stackoverflow(query = geterrmessage(), rlang = TRUE)
```

```
search_so(query = geterrmessage(), rlang = TRUE)
```

```
search_github(query = geterrmessage(), rlang = TRUE)
```

```
search_gh(query = geterrmessage(), rlang = TRUE)
```

```
search_bitbucket(query = geterrmessage(), rlang = TRUE)
```

```
search_bb(query = geterrmessage(), rlang = TRUE)
```

Arguments

query	Contents of string to search. Default is the error message.
site	Name of site to search on. Supported options: "google" (default), "bing", "duckduckgo", "startpage", "rstudio community", "twitter", "stackoverflow", "github", and "bitbucket".
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

Google Search

The `search_google` function searches **Google** using: <https://google.com/search?q=<query>>

See <https://moz.com/blog/the-ultimate-guide-to-the-google-search-parameters> for details.

Bing Search

The `search_bing()` function searches **Bing** using: <https://bing.com/search?q=<query>>

DuckDuckGo Search

The `search_duckduckgo()` and `search_ddg()` functions both search **DuckDuckGo** using: <https://duckduckgo.com/?q=<query>>

Startpage Search

The `search_startpage()` function searches **startpage** using: <https://startpage.com/do/dsearch?query=<query>>

For additional details regarding **startpage**'s search interface please see: <https://support.startpage.com/index.php?Knowledgebase/Article/View/1261/0/add-familystartpagecom-as-the-default-search-engine>

RStudio Community Search

The `search_rstudio_community()` and `search_rscom()` functions both search **RStudio Community** using: <https://community.rstudio.com/search?q=<query>>

For additional details regarding **RStudio Community**'s search interface please see the **Discourse** API documentation: <https://docs.discourse.org/#tag/Search>

Twitter Search

The `search_twitter()` functions search **Twitter** using: <https://twitter.com/search?q=<query>>

For additional details regarding **Twitter**'s search interface please see: <https://help.twitter.com/en/using-twitter/twitter-advanced-search>

StackOverflow Search

The `search_stackoverflow()` and `search_so()` functions both search **StackOverflow** using: <https://stackoverflow.com>

For additional details regarding **StackOverflow**'s search interface please see: <https://stackoverflow.com/help/advanced-search-parameters-jobs>

GitHub Search

The `search_github()` and `search_gh()` functions both search **GitHub** using: <https://github.com/search?q=<query>+1>

For additional details regarding **GitHub**'s search interface please see: <https://help.github.com/categories/searching-for-information-on-github/> and <https://help.github.com/articles/searching-code/>

BitBucket Search

The `search_bitbucket()` and `search_bb()` functions both search **BitBucket** using: <https://bitbucket.com/search?q=1>

For additional details regarding **BitBucket**'s search interface please see: <https://confluence.atlassian.com/bitbucket/code-search-in-bitbucket-873876782.html>

See Also

[search_google\(\)](#), [search_bing\(\)](#), [search_duckduckgo\(\)](#), [search_startpage\(\)](#), [search_twitter\(\)](#), [search_rstudio_community\(\)](#), [search_stackoverflow\(\)](#), [search_github\(\)](#), [search_bitbucket\(\)](#), and [searcher\(\)](#)

Examples

```
# Search in a generic way
search_site("r-project", "google")

# Search Google
search_google("r-project")

# Search Bing
search_bing("Microsoft R")

# Search DuckDuckGo
search_duckduckgo("R language")

# Search startpage
search_startpage("VS Code")

# Search RStudio Community
search_rstudio_community("RStudio IDE")

# Search Twitter
search_twitter("searcher")

# Search StackOverflow for Convolutions in the r tag
search_stackoverflow("convolutions")
```

```
# Search all languages on StackOverflow for convolutions
search_stackoverflow("convolutions", rlang = FALSE)

# Search GitHub Issues for bivariate normal in the language:r
search_github("bivariate normal")

# Search all languages on GitHub Issues for bivariate normal
search_github("bivariate normal", rlang = FALSE)

# Search BitBucket for assertions
search_bitbucket("assertions")

## Not run:
# On error, automatically search the message on google
options(error = searcher("google"))
options(error = search_google)

## End(Not run)
```

Index

`_PACKAGE` (searcher-package), 2

`options()`, 2

`search_bb` (search_site), 4

`search_bing` (search_site), 4

`search_bing()`, 6

`search_bitbucket` (search_site), 4

`search_bitbucket()`, 6

`search_ddg` (search_site), 4

`search_duckduckgo` (search_site), 4

`search_duckduckgo()`, 6

`search_gh` (search_site), 4

`search_github` (search_site), 4

`search_github()`, 6

`search_google` (search_site), 4

`search_google()`, 6

`search_ixquick` (search_site), 4

`search_ixquick()`, 3

`search_rscom` (search_site), 4

`search_rstudio_community` (search_site),
4

`search_rstudio_community()`, 6

`search_site`, 4

`search_so` (search_site), 4

`search_sp` (search_site), 4

`search_stackoverflow` (search_site), 4

`search_stackoverflow()`, 6

`search_startpage` (search_site), 4

`search_startpage()`, 3, 6

`search_twitter` (search_site), 4

`search_twitter()`, 6

`searcher`, 2

`searcher()`, 6

`searcher-defunct`, 3

`searcher-package`, 2