

Package: searcher (via r-universe)

September 19, 2026

Title Query Search Interfaces and AI Assistants

Version 0.1.0

Description Provides search interfaces to look up terms on major search engines including 'Google', 'Bing', 'DuckDuckGo', 'Startpage', 'Ecosia', 'Brave', 'Kagi', 'rseek', social platforms like 'X' (formerly Twitter), 'BlueSky', 'Mastodon', programming communities such as 'StackOverflow', 'Posit Community', and code repositories including 'GitHub', 'grep.app', and 'BitBucket'. Additionally, provides direct integration with AI assistants through specialized query functions for 'ChatGPT', 'Claude AI', 'Perplexity AI', 'Mistral AI', 'Microsoft Copilot', and 'Meta AI', complete with an AI prompt management system for R-optimized interactions. Upon searching or querying, a browser window will open with the search results or AI interface pre-populated with the specified query.

URL <https://github.com/coatless-rpkg/searcher>,
<https://r-pkg.thecoatlessprofessor.com/searcher/>

BugReports <https://github.com/coatless-rpkg/searcher/issues>

Depends R (>= 3.3.0)

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

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

VignetteBuilder knitr

Collate 'ai-prompts.R' 'index-sites.R' 'ai-search-functions.R'
'defunct-functions.R' 'search-functions.R' 'searcher-package.R'
'utilities.R'

Repository <https://coatless-rpkg.r-universe.dev>

Date/Publication 2025-05-27 19:16:51 UTC

RemoteUrl <https://github.com/coatless-rpkg/searcher>

RemoteRef HEAD

RemoteSha 250c51a338e555b35b31ca98dd3176b81d62f0cf

Contents

searcher-package	3
ai_prompt	4
ai_prompt_active	5
ai_prompt_clear	5
ai_prompt_list	6
ai_prompt_management	6
ai_prompt_register	7
ai_prompt_remove	7
ask_bing_copilot	8
ask_chatgpt	9
ask_claude	10
ask_grok	11
ask_meta_ai	11
ask_mistral	12
ask_perplexity	13
search_bing	14
search_bitbucket	15
search_bluesky	16
search_brave	16
search_duckduckgo	17
search_ecosia	18
search_github	18
search_google	19
search_grep	20
search_kagi	21
search_mastodon	21
search_posit_community	22
search_qwant	23
search_rseek	24
search_site	24
search_stackoverflow	25
search_startpage	26
search_twitter	27
searcher	28
searcher-defunct	29

Index

30

Description

Provides search interfaces to look up terms on major search engines including 'Google', 'Bing', 'DuckDuckGo', 'Startpage', 'Ecosia', 'Brave', 'Kagi', 'rseek', social platforms like 'X' (formerly Twitter), 'BlueSky', 'Mastodon', programming communities such as 'StackOverflow', 'Posit Community', and code repositories including 'GitHub', 'grep.app', and 'BitBucket'. Additionally, provides direct integration with AI assistants through specialized query functions for 'ChatGPT', 'Claude AI', 'Perplexity AI', 'Mistral AI', 'Microsoft Copilot', and 'Meta AI', complete with an AI prompt management system for R-optimized interactions. Upon searching or querying, a browser window will open with the search results or AI interface pre-populated with the specified query.

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".
- Other options are used to set the default prompts for AI services
 - `searcher.chatgpt_prompt`: Default prompt for OpenAI's ChatGPT.
 - `searcher.claude_prompt`: Default prompt for Anthropic's Claude AI.
 - `searcher.perplexity_prompt`: Default prompt for Perplexity AI.
 - `searcher.mistral_prompt`: Default prompt for Mistral AI's Le Chat.
 - `searcher.bing_copilot_prompt`: Default prompt for Microsoft's (Bing) Copilot.
 - `searcher.grok_prompt`: Default prompt for xAI's Grok AI.
 - `searcher.meta_ai_prompt`: Default prompt for Meta's AI.

Author(s)

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

Other contributors:

- Alex Rossell Hayes <alexander@rossellhayes.com> ([ORCID](#)) [contributor]

See Also

Useful links:

- <https://github.com/coatless-rpkg/searcher>
- <https://r-pkg.thecoatlessprofessor.com/searcher/>
- Report bugs at <https://github.com/coatless-rpkg/searcher/issues>

ai_prompt

Set or View Active System-level AI Prompt

Description

Sets a system-level prompt to be used with all AI search functions. When called with no arguments, returns the currently active prompt.

Usage

```
ai_prompt(prompt_name = NULL)
```

Arguments

prompt_name	Name of a prompt from the prompt library, or a custom prompt text. Use <code>ai_prompt_list()</code> to see available prompt names. If <code>NULL</code> , returns the current active prompt without changing it.
-------------	---

Value

Invisibly returns the active prompt text. If called without arguments, returns the active prompt visibly.

Examples

```
## Not run:
# Set a predefined prompt
ai_prompt("debugging")

# Set a custom prompt
ai_prompt("Explain this R error in simple terms with examples:")

# Check current active prompt
ai_prompt()

# Clear the system prompt
ai_prompt(NULL)

## End(Not run)
```

ai_prompt_active	<i>Get Currently Active System Prompt</i>
------------------	---

Description

Returns the currently active system-level prompt, if any.

Usage

```
ai_prompt_active()
```

Value

The active prompt text, or NULL if no system prompt is set.

Examples

```
## Not run:  
# Check current active prompt  
ai_prompt_active()  
  
## End(Not run)
```

ai_prompt_clear	<i>Clear the Active System Prompt</i>
-----------------	---------------------------------------

Description

Clears the currently active system-level prompt.

Usage

```
ai_prompt_clear()
```

Value

Invisibly returns NULL.

Examples

```
## Not run:  
# Clear the system prompt  
ai_prompt_clear()  
  
## End(Not run)
```

ai_prompt_list	<i>List Available AI Prompts</i>
----------------	----------------------------------

Description

Lists all available prompts in the prompt library.

Usage

```
ai_prompt_list()
```

Value

A named list of available prompts.

Examples

```
## Not run:  
# List all available prompts  
ai_prompt_list()  
  
## End(Not run)
```

ai_prompt_management	<i>AI Prompt Management System</i>
----------------------	------------------------------------

Description

A set of functions to manage and apply prompts for AI search services. These functions allow you to create, maintain, and use a library of effective prompts for different AI assistants and scenarios.

Details

The prompt management system works with multiple levels of prompts:

1. System-level prompt: Set with `ai_prompt()`, applies across all AI services
2. Service-specific prompts: Set with `options()` or in function calls
3. Default prompts: Built-in prompts that ship with the package

When a search is performed, prompts are applied in this order, with the query at the end.

ai_prompt_register	<i>Register a New AI Prompt</i>
--------------------	---------------------------------

Description

Adds a new prompt to the prompt library.

Usage

```
ai_prompt_register(name, prompt_text, overwrite = FALSE)
```

Arguments

name	Name for the new prompt.
prompt_text	The prompt text to register.
overwrite	Whether to overwrite an existing prompt with the same name. Default is FALSE.

Value

Invisibly returns the updated prompt library.

Examples

```
## Not run:  
# Register a new prompt  
ai_prompt_register(  
  "tidyverse",  
  paste("As a tidyverse expert, explain how to solve this problem using",  
        "dplyr, tidyr, and other tidyverse packages:"  
  )  
)  
## End(Not run)
```

ai_prompt_remove	<i>Remove an AI Prompt from the Library</i>
------------------	---

Description

Removes a prompt from the prompt library.

Usage

```
ai_prompt_remove(name)
```

Arguments

name Name of the prompt to remove.

Value

Invisibly returns the updated prompt library.

Examples

```
## Not run:
# Remove a prompt
ai_prompt_remove("tidyverse")

## End(Not run)
```

ask_bing_copilot	<i>Search with Bing Copilot</i>
------------------	---------------------------------

Description

Searches Microsoft Bing Copilot, which combines web search results with AI-generated responses. This makes it useful for queries that benefit from current web information.

Usage

```
ask_bing_copilot(query = geterrmessage(), prompt = NULL)

ask_copilot(query = geterrmessage(), prompt = NULL)
```

Arguments

query Contents of string to send to AI Service. Default is the last error message.

prompt Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_chatgpt\(\)](#), [ask_claude\(\)](#), [ask_grok\(\)](#), [ask_meta_ai\(\)](#), [ask_mistral\(\)](#), [ask_perplexity\(\)](#)

Examples

```
# Basic query
ask_bing_copilot("Latest R package for geospatial analysis")

# Using a custom prompt
ask_bing_copilot("Write a function to calculate the median",
  prompt = "Show multiple approaches:")
```

ask_chatgpt

*Search with ChatGPT***Description**

Opens a browser with OpenAI's ChatGPT interface and your query. This function allows you to ask questions, get code help, or search for information using ChatGPT.

Usage

```
ask_chatgpt(query = geterrmessage(), prompt = NULL)
```

Arguments

query	Contents of string to send to AI Service. Default is the last error message.
prompt	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_claude\(\)](#), [ask_grok\(\)](#), [ask_meta_ai\(\)](#), [ask_mistral\(\)](#), [ask_perplexity\(\)](#)

Examples

```
# Basic query
ask_chatgpt("How to join two dataframes in R?")

# Using a custom prompt
ask_chatgpt("Error: object 'mtcars' not found",
  prompt = "Debug this error step by step:")

# Searching the last error
## Not run:
tryCatch(
  median("not a number"),
```

```

    error = function(e) ask_chatgpt()
  )

  ## End(Not run)

```

ask_claude

Search with Claude

Description

Opens Anthropic's Claude AI assistant with your query. Claude can provide thorough answers to complex questions and offers excellent code explanations.

Usage

```
ask_claude(query = geterrmessage(), prompt = NULL)
```

Arguments

query	Contents of string to send to AI Service. Default is the last error message.
prompt	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_chatgpt\(\)](#), [ask_grok\(\)](#), [ask_meta_ai\(\)](#), [ask_mistral\(\)](#), [ask_perplexity\(\)](#)

Examples

```

# Basic query
ask_claude("Explain what purrr::map_df does")

# Using a custom prompt
ask_claude("Compare tidyr::pivot_wider vs tidyr::spread",
  prompt = "Provide examples of when to use each:")

```

`ask_grok`*Search with Grok*

Description

Searches xAI's Grok, which provides AI assistance focused on maximize truth and objectivity.

Usage

```
ask_grok(query = geterrmessage(), prompt = NULL)
```

```
ask_xai(query = geterrmessage(), prompt = NULL)
```

Arguments

<code>query</code>	Contents of string to send to AI Service. Default is the last error message.
<code>prompt</code>	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_chatgpt\(\)](#), [ask_claude\(\)](#), [ask_meta_ai\(\)](#), [ask_mistral\(\)](#), [ask_perplexity\(\)](#)

Examples

```
# Basic query
ask_grok("What are the best practices for R package development?")

# Using a custom prompt
ask_grok("How to optimize this R code for performance?",
         prompt = "Focus on efficiency and best practices:")
```

`ask_meta_ai`*Search with Meta AI*

Description

Searches Meta AI, which provides general-purpose AI assistance with a focus on conversational responses.

Usage

```
ask_meta_ai(query = geterrmessage(), prompt = NULL)
```

Arguments

query	Contents of string to send to AI Service. Default is the last error message.
prompt	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_chatgpt\(\)](#), [ask_claude\(\)](#), [ask_grok\(\)](#), [ask_mistral\(\)](#), [ask_perplexity\(\)](#)

Examples

```
# Basic query
ask_meta_ai("What are the best R packages for visualization?")

# Using a custom prompt
ask_meta_ai("How to create a heatmap in R",
            prompt = "Compare ggplot2 and base R approaches:")
```

ask_mistral	<i>Search with Mistral AI</i>
-------------	-------------------------------

Description

Launches Mistral AI with your query. Mistral is known for its strong reasoning capabilities and efficiency.

Usage

```
ask_mistral(query = geterrmessage(), prompt = NULL)

ask_le_chat(query = geterrmessage(), prompt = NULL)
```

Arguments

query	Contents of string to send to AI Service. Default is the last error message.
prompt	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_chatgpt\(\)](#), [ask_claude\(\)](#), [ask_grok\(\)](#), [ask_meta_ai\(\)](#), [ask_perplexity\(\)](#)

Examples

```
# Basic query
ask_mistral("How to handle missing data in R?")

# Using a custom prompt
ask_mistral("Fix this code: ggplot(mtcars, aes(x=mpg, y=hp) + geom_point()",
           prompt = "Explain the error and fix it:")
```

ask_perplexity	<i>Search with Perplexity</i>
----------------	-------------------------------

Description

Searches with Perplexity AI, which provides answers with citations to sources. This makes it particularly useful for research-oriented queries.

Usage

```
ask_perplexity(query = geterrmessage(), prompt = NULL)
```

Arguments

query	Contents of string to send to AI Service. Default is the last error message.
prompt	Optional prompt prefix to add before your query to guide how the AI Service responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

See Also

Other AI assistants: [ask_bing_copilot\(\)](#), [ask_chatgpt\(\)](#), [ask_claude\(\)](#), [ask_grok\(\)](#), [ask_meta_ai\(\)](#), [ask_mistral\(\)](#)

Examples

```
# Basic query
ask_perplexity("Compare dplyr vs data.table")

# Using a custom prompt
ask_perplexity("Best packages for time series in R",
               prompt = "Provide citations and compare performance:")
```

search_bing

Search Bing

Description

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

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Bing
search_bing("Microsoft R")
```

search_bitbucket	<i>Search BitBucket</i>
------------------	-------------------------

Description

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

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Details

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

Value

The generated search URL or an empty string.

See Also

Other code repositories: [search_github\(\)](#), [search_grep\(\)](#)

Examples

```
# Search BitBucket for assertions
search_bitbucket("assertions")

# Search all languages on BitBucket for assertions
search_bitbucket("assertions", rlang = FALSE)
```

search_bluesky	<i>Search BlueSky</i>
----------------	-----------------------

Description

The `search_bluesky()` functions search BlueSky using: `https://bsky.app/search?q=%23stats+<query>`

Usage

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other community sites: [search_mastodon\(\)](#), [search_posit_community\(\)](#), [search_stackoverflow\(\)](#), [search_twitter\(\)](#)

Examples

```
# Search BlueSky
search_bluesky("searcher")
```

search_brave	<i>Search Brave</i>
--------------	---------------------

Description

The `search_brave()` function searches Brave using: `https://search.brave.com/search?q=<query>&source=web`

Usage

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Brave
search_brave("webR")
```

search_duckduckgo	<i>Search DuckDuckGo</i>
-------------------	--------------------------

Description

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

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search DuckDuckGo
search_duckduckgo("R language")
```

search_ecosia	<i>Search Ecosia</i>
---------------	----------------------

Description

The `search_ecosia()` function searches Ecosia using: <https://www.ecosia.org/search?q=<query>>

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Details

For additional details regarding Ecosia's search interface please see: <https://support.ecosia.org/article/657-installing-ecosia-on-your-desktop-device>

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Ecosia
search_ecosia("climate change R analysis")
```

search_github	<i>Search GitHub</i>
---------------	----------------------

Description

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

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Details

For additional details regarding **GitHub**'s search interface please see: <https://docs.github.com/en/enterprise-cloud@latest/search-github/getting-started-with-searching-on-github/about-searching-on-github> and <https://docs.github.com/en/search-github/searching-on-github/searching-code/>

Value

The generated search URL or an empty string.

See Also

Other code repositories: [search_bitbucket\(\)](#), [search_grep\(\)](#)

Examples

```
# 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_google	<i>Search Google</i>
---------------	----------------------

Description

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

Usage

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Google
search_google("r-project")

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

## End(Not run)
```

search_grep

Search Grep.app

Description

The `search_grep()` function searches all public code on [GitHub](#) using [grep.app](#) by querying: [https://grep.app/search?q=<query-here>&filter\[lang\]\[0\]=R](https://grep.app/search?q=<query-here>&filter[lang][0]=R)

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other code repositories: [search_bitbucket\(\)](#), [search_github\(\)](#)

Examples

```
# Search R code on GitHub for numerical optimization
search_grep("optim")

# Search all code on GitHub for numerical optimization
search_grep("optim", rlang = FALSE)
```

`search_kagi`*Search Kagi*

Description

The `search_kagi()` function searches Kagi using: `https://kagi.com/search?q=<query>`

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Details

This is a paid search engine, and you will need to sign up for an account to use it.

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Kagi
search_kagi("advanced R programming")
```

`search_mastodon`*Search Mastodon*

Description

The `search_mastodon()` functions search Mastodon using: `https://mastodon.social/search?q=%23rstats+<query>`

Usage

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other community sites: [search_bluesky\(\)](#), [search_posit_community\(\)](#), [search_stackoverflow\(\)](#), [search_twitter\(\)](#)

Examples

```
# Search Mastodon
search_mastodon("searcher")
```

```
search_posit_community
      Search Posit Community
```

Description

The `search_posit_community()` and `search_posit()` functions both search **Posit Community** using: `https://forum.posit.co/search?q=<query>`

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Details

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

Value

The generated search URL or an empty string.

See Also

Other community sites: [search_bluesky\(\)](#), [search_mastodon\(\)](#), [search_stackoverflow\(\)](#), [search_twitter\(\)](#)

Examples

```
# Search Posit Community
search_posit_community("RStudio IDE")
```

search_qwant	<i>Search Qwant</i>
--------------	---------------------

Description

The `search_qwant()` function searches Qwant using: `https://www.qwant.com/?q=<query>`

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_rseek\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Qwant
search_qwant("Quarto")
```

search_rseek	<i>Search Rseek</i>
--------------	---------------------

Description

The `search_rseek()` function searches **Rseek** using: `https://rseek.org/?q=<query>`

Usage

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

Arguments

<code>query</code>	Contents of string to search. Default is the error message.
<code>rlang</code>	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_startpage\(\)](#)

Examples

```
# Search Rseek
search_rseek("searcher")
```

search_site	<i>Search a Query on a Search Portal in a Web Browser</i>
-------------	---

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", "qwant", "rseek",  
    "brave", "kagi", "posit community", "posit", "twitter", "x", "bluesky", "mastodon",  
    "stackoverflow", "so", "github", "gh", "grep", "bitbucket", "bb", "chatgpt",  
    "claude", "perplexity", "mistral", "bing copilot", "copilot", "grok", "meta ai",  
    "meta"),  
  rlang = TRUE,  
  prompt = NULL  
)
```

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" or "ddg", "startpage" (formerly "ixquick") or "sp", "qwant", "rseek", "brave", "kagi", "posit community" (formerly "rstudio community") or "posit", "twitter" or "x", "bluesky", "mastodon", "stackoverflow", "github", "grep", "bitbucket", "chatgpt", "claude", "perplexity", "mistral" or "le chat", "bing copilot" or "copilot", and "grok" or "xai", "meta ai" or "meta".
rlang	Search for results written in R. Default is TRUE
prompt	Optional prompt prefix to add before your query to guide how the AI responds. If NULL, uses the service-specific default prompt option.

Value

The generated search URL or an empty string.

Examples

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

search_stackoverflow	<i>Search StackOverflow</i>
----------------------	-----------------------------

Description

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

Usage

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Details

For additional details regarding [StackOverflow](#)'s search interface please see:

<https://stackoverflow.com/help/searching>

Value

The generated search URL or an empty string.

See Also

Other community sites: [search_bluesky\(\)](#), [search_mastodon\(\)](#), [search_posit_community\(\)](#), [search_twitter\(\)](#)

Examples

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

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

search_startpage	<i>Search Startpage</i>
------------------	-------------------------

Description

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

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Value

The generated search URL or an empty string.

See Also

Other search engines: [search_bing\(\)](#), [search_brave\(\)](#), [search_duckduckgo\(\)](#), [search_ecosia\(\)](#), [search_google\(\)](#), [search_kagi\(\)](#), [search_qwant\(\)](#), [search_rseek\(\)](#)

Examples

```
# Search startpage
search_startpage("VS Code")
```

search_twitter	<i>Search Twitter/X</i>
----------------	-------------------------

Description

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

Usage

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

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

Arguments

query	Contents of string to search. Default is the error message.
rlang	Search for results written in R. Default is TRUE

Details

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

Value

The generated search URL or an empty string.

See Also

Other community sites: [search_bluesky\(\)](#), [search_mastodon\(\)](#), [search_posit_community\(\)](#), [search_stackoverflow\(\)](#)

Examples

```
# Search Twitter
search_twitter("searcher")
```

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" or "ddg", "startpage" (formerly "ixquick") or "sp", "qwant", "rseek", "brave", "kagi", "posit community" (formerly "rstudio community") or "posit", "twitter" or "x", "bluesky", "mastodon", "stackoverflow", "github", "grep", "bitbucket", "chatgpt", "claude", "perplexity", "mistral" or "le chat", "bing copilot" or "copilot", and "grok" or "xai", "meta ai" or "meta".
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	<i>Defunct functions in searcher</i>
------------------	--------------------------------------

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_rstudio_community()`: The function binding was removed as the company name changed to "Posit". Please use [search_posit_community\(\)](#).
- `search_rscom()`: The function binding was removed as the company name changed to "Posit". Please use [search_posit\(\)](#).

Index

* AI assistants

- ask_bing_copilot, 8
- ask_chatgpt, 9
- ask_claude, 10
- ask_grok, 11
- ask_meta_ai, 11
- ask_mistral, 12
- ask_perplexity, 13

* code repositories

- search_bitbucket, 15
- search_github, 18
- search_grep, 20

* community sites

- search_bluesky, 16
- search_mastodon, 21
- search_posit_community, 22
- search_stackoverflow, 25
- search_twitter, 27

* search engines

- search_bing, 14
- search_brave, 16
- search_duckduckgo, 17
- search_ecosia, 18
- search_google, 19
- search_kagi, 21
- search_qwant, 23
- search_rseek, 24
- search_startpage, 26

ai_prompt, 4

ai_prompt_active, 5

ai_prompt_clear, 5

ai_prompt_list, 6

ai_prompt_management, 6

ai_prompt_register, 7

ai_prompt_remove, 7

ask_bing_copilot, 8, 9–13

ask_chatgpt, 8, 9, 10–13

ask_claude, 8, 9, 10, 11–13

ask_copilot (ask_bing_copilot), 8

ask_grok, 8–10, 11, 12, 13

ask_le_chat (ask_mistral), 12

ask_meta_ai, 8–10, 11, 11, 13

ask_mistral, 8–11, 12, 12, 13

ask_perplexity, 8–12, 13, 13

ask_xai (ask_grok), 11

options(), 3

search_bb (search_bitbucket), 15

search_bing, 14, 17, 18, 20, 21, 23, 24, 27

search_bitbucket, 15, 19, 20

search_bluesky, 16, 22, 23, 26, 27

search_brave, 14, 16, 17, 18, 20, 21, 23, 24, 27

search_ddg (search_duckduckgo), 17

search_duckduckgo, 14, 17, 17, 18, 20, 21, 23, 24, 27

search_ecosia, 14, 17, 18, 20, 21, 23, 24, 27

search_gh (search_github), 18

search_github, 15, 18, 20

search_google, 14, 17, 18, 19, 21, 23, 24, 27

search_grep, 15, 19, 20

search_kagi, 14, 17, 18, 20, 21, 23, 24, 27

search_mastodon, 16, 21, 23, 26, 27

search_posit (search_posit_community), 22

search_posit(), 29

search_posit_community, 16, 22, 22, 26, 27

search_posit_community(), 29

search_qwant, 14, 17, 18, 20, 21, 23, 24, 27

search_rseek, 14, 17, 18, 20, 21, 23, 24, 27

search_site, 24

search_so (search_stackoverflow), 25

search_sp (search_startpage), 26

search_stackoverflow, 16, 22, 23, 25, 27

search_startpage, 14, 17, 18, 20, 21, 23, 24, 26

search_startpage(), 29

search_twitter, 16, 22, 23, 26, 27

`search_x (search_twitter)`, [27](#)
`searcher`, [28](#)
`searcher-defunct`, [29](#)
`searcher-package`, [3](#)