Package 'pkgsearch'

May 9, 2026

Title Search and Query CRAN R Packages

Version 3.1.5

Description Search CRAN metadata about packages by keyword, popularity, recent activity, package name and more. Uses the 'R-hub' search server, see <https://r-pkg.org> and the CRAN metadata database, that contains information about CRAN packages. Note that this is _not_ a CRAN project.

License MIT + file LICENSE

URL https://github.com/r-hub/pkgsearch, https://r-hub.github.io/pkgsearch/

BugReports https://github.com/r-hub/pkgsearch/issues

Imports curl, jsonlite

Suggests covr, memoise, mockery, pillar, pingr (>= 2.0.0), rstudioapi, shiny, shinyjs, shinyWidgets, testthat (>= 3.0.0), whoami, withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1.9000

NeedsCompilation no

Author Gábor Csárdi [aut, cre], Maëlle Salmon [aut] (ORCID: <https://orcid.org/0000-0002-2815-0399>), R Consortium [fnd]

Maintainer Gábor Csárdi <csardi.gabor@gmail.com>

Repository CRAN

Date/Publication 2025-04-12 09:40:02 UTC

Contents

`advanced_search` *Advanced CRAN package search*

Description

See the Elastic documentation for the syntax and features: https://www.elastic.co/guide/en/elasticsearch/reference/current/que
dsl-query-string-query.html

Usage

```
advanced_search(
  ...,
  json = NULL,
  format = c("short", "long"),
  from = 1,
  size = 10
)
```

Arguments

`...`	Search terms. For named terms, the name specifies the field to search for. For unnamed ones, the term is taken as is. The individual terms are combined with the `AND` operator.
`json`	A character string that contains the query to send to Elastic. If this is not `NULL`, then you cannot specify any search terms in `...`.
`format`	Default formatting of the results. *short* only outputs the name and title of the packages, *long* also prints the author, last version, full description and URLs. Note that this only affects the default printing, and you can still inspect the full results, even if you specify *short* here.
`from`	Where to start listing the results, for pagination.
`size`	The number of results to list.

Value

Search hits.

Examples

```
# All orphaned packages
advanced_search(Maintainer = "ORPHANED")

# Packages with both Hester and Wickham as authors
advanced_search(Author = "Hester", Author = "Wickham")
advanced_search("Author: Hester AND Author: Wickham")

# Packages with Hester but not Wickham as author
advanced_search(Author = "Hester AND NOT Wickham")

# Packages with Hester as an Author, and Wickham in any field
advanced_search(Author = "Hester", "Wickham")

# Packages with Hester as an Author and Wickham nowhere in the metadata
advanced_search(Author = "Hester", "NOT Wickham")

# Packages for permutation tests and permissive licenses
advanced_search("permutation test AND NOT License: GPL OR GNU")

# Packages that have a certain field
advanced_search("_exists_" = "URL")

# Packages that do not have a certain field:
advanced_search("NOT _exists_: URL")

# The same but as JSON query
query <- '{
"query": {
  "bool": {
    "must_not": {
      "exists": {
        "field": "URL"
      }
    }
  }
}
}'
advanced_search(json = query)

# Regular expressions
advanced_search(Author = "/Joh?nathan/")

# Fuzzy search
advanced_search(Author = "Johnathan~1")
```

`cran_events` *List of all CRAN events (new, updated, archived packages)*

Description

List of all CRAN events (new, updated, archived packages)

Usage

```
cran_events(releases = TRUE, archivals = TRUE, limit = 10, from = 1)

## S3 method for class 'cran_event_list'
summary(object, ...)

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

Arguments

releases	Whether to include package releases.
archivals	Whether to include package archivals.
limit	Number of events to list.
from	Where to start the list, for pagination.
object	Object to summarize.
...	Additional arguments are ignored currently.
x	Object to print.

Value

List of events.

Examples

```
cran_events()
cran_events(limit = 5, releases = FALSE)
cran_events(limit = 5, archivals = FALSE)
summary(cran_events(limit = 10))
```

---

cran_new *New CRAN packages*

Description

List the latest new CRAN packages.

Usage

```
cran_new(from = "last-week", to = "now", last = Inf)
```

Arguments

`from`	Start of the time interval to query. Possible values: • `"last-week"` • `"last-month"` • A Date object to be used as a start date. • A POSIXt object to be used as the start date. • A difftime object to used as the time interval until now. • An integer scalar, the number of days until today. • A character string that is converted to a start date using `as.POSIXct()`.
`to`	End of the time interval to query. It accepts the same kinds of values as `from`, and additionally it can also be the string `"now"`, to specify the current date and time.
`last`	Integer to limit the number of returned packages.

Value

Data frame of package descriptions.

Examples

```
# Last week
cran_new("last-week")

# Last month
cran_new("last-month")

# Last 5 days
cran_new(from = 5)

# From a given date, but at most 10
cran_new(from = "2021-04-06", last = 10)

# March of 2021
cran_new(from = "2021-03-01", to = "2021-04-01")
```

`cran_package` *Metadata about a CRAN package*

Description

Metadata about a CRAN package

Usage

```
cran_package(name, version = NULL)
```

Arguments

name	Name of the package.
version	The package version to query. If NULL, the latest version if returned.

Value

The package metadata, in a named list.

Examples

```
cran_package("pkgsearch")
```

---

cran_packages *Metadata about multiple CRAN packages*

Description

Metadata about multiple CRAN packages

Usage

```
cran_packages(names)
```

Arguments

names	Package names. May also contain versions, separated by a @ character.

Value

A data frame of package metadata, one package per row.

Examples

```
# Get metadata about one package
cran_packages("rhub")
# Get metadata about two packages
cran_packages(c("rhub", "testthat"))
# Get metadata about two packages at given versions
cran_packages(c("rhub@1.1.1", "testthat@2.2.1", "testthat@2.2.0"))
# If a version does not exist nothing is returned
cran_packages("rhub@notaversion")
```

cran_package_history *Query the history of a package*

Description

Query the history of a package

Usage

```
cran_package_history(package)
```

Arguments

package	Package name.

Value

A data frame, with one row per package version.

Examples

```
cran_package_history("igraph")
```

cran_top_downloaded *Top downloaded packages*

Description

Last week.

Usage

```
cran_top_downloaded()
```

Details

You can use the cranlogs package to get more flexibility into what is returned.

Value

Data frame of top downloaded packages.

Examples

```
cran_top_downloaded()
```

cran_trending *Trending R packages*

Description

Trending packages are the ones that were downloaded at least 1000 times during last week, and that substantially increased their download counts, compared to the average weekly downloads in the previous 24 weeks. The percentage of increase is also shown in the output.

Usage

```
cran_trending()
```

Value

Data frame of trending packages.

Examples

```
cran_trending()
```

pkg_search *Search CRAN packages*

Description

`pkg_search()` starts a new search query, or shows the details of the previous query, if called without arguments.

`ps()` is an alias to `pkg_search()`.

`more()` retrieves that next page of results for the previous query.

Usage

```
pkg_search(query = NULL, format = c("short", "long"), from = 1, size = 10)

ps(query = NULL, format = c("short", "long"), from = 1, size = 10)

more(format = NULL, size = NULL)

## S3 method for class 'pkg_search_result'
summary(object, ...)

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

Arguments

query	Search query string. If this argument is missing or NULL, then the results of the last query are printed, in *short* and *long* formats, in turns for successive pkg_search() calls. If this argument is missing, then all other arguments are ignored.
format	Default formatting of the results. *short* only outputs the name and title of the packages, *long* also prints the author, last version, full description and URLs. Note that this only affects the default printing, and you can still inspect the full results, even if you specify *short* here.
from	Where to start listing the results, for pagination.
size	The number of results to list.
object	Object to summarize.
...	Additional arguments, ignored currently.
x	Object to print.

Details

Note that the search needs a working Internet connection.

Value

A data frame with columns:

- score: Score of the hit. See Section *Scoring* for some details.
- package: Package name.
- version: Latest package version.
- title: Package title.
- description: Short package description.
- date: Time stamp of the last release.
- maintainer_name: Name of the package maintainer.
- maintainer_email: Email address of the package maintainer.
- revdeps: Number of (strong and weak) reverse dependencies of the package.
- downloads_last_month: Raw number of package downloads last month, from the RStudio CRAN mirror.
- license: Package license.
- url: Package URL(s).
- bugreports: URL of issue tracker, or email address for bug reports.

Examples

```
# Example
ps("survival")

# Pagination
ps("networks")
more()

# Details
ps("visualization")
ps()

# See the underlying data frame
ps("ropensci")
ps()[]
```

`pkg_search_addin` *RStudio addin to search CRAN packages*

Description

Call this function from RStudio for best results. You can also use it without RStudio, then it will run in the web browser.

Usage

```
pkg_search_addin(query = "", viewer = c("dialog", "browser"))
```

Arguments

`query`	Query string to start the addin with.
`viewer`	Whether to show the addin within RStudio (`"dialog"`), or in a web browser (`"browser"`).

Details

The app has:

- A search tab for free text search, very much like the `pkg_search()` function.
- The list of recently updated packages.
- The list of top packages: most downloaded, most depended upon, and trending packages.
- Package list by maintainer.

Examples

```
pkg_search_addin()

# Start with a search query
pkg_search_addin("permutation test")
```

Index