Package ‘available’

October 12, 2022

Title  Check if the Title of a Package is Available, Appropriate and Interesting

Version  1.1.0

Description  Check if a given package name is available to use. It checks the name's validity. Checks if it is used on 'GitHub', 'CRAN' and 'Bioconductor'. Checks for unintended meanings by querying 'Wiktionary' and Wikipedia.

Depends  R (>= 3.3.0)

License  MIT + file LICENSE

Encoding  UTF-8

RoxygenNote  7.2.0.9000

Imports  cli, clisymbols, crayon, tidytext, desc, glue, jsonlite, memoise, SnowballC, stringdist, tibble, yesno

Suggests  usethis, BiocManager, testthat, covr

URL  https://github.com/r-lib/available

BugReports  https://github.com/r-lib/available/issues

NeedsCompilation  no

Author  Carl Ganz [aut],
Gábor Csárdi [aut, cre],
Jim Hester [aut],
Molly Lewis [aut],
Rachael Tatman [aut]

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

Repository  CRAN

Date/Publication  2022-07-10 19:00:02 UTC

R topics documented:

available ................................................................. 2
available_on_bioc .................................................. 3
available_on_github .................................................. 3
available

Description
Searches performed

- Valid package name
- Already taken on CRAN
- Positive or negative sentiment

Usage
available(name, browse = getOption("available.browse", TRUE), ...)

Arguments

name       Name of package to search
browse     Whether browser should be opened for all web links, default = TRUE. Default can be changed by setting available.browse in .Rprofile. See Startup for more details.
...

Additional arguments passed to utils::available.packages().

Examples

## Not run:
# Check if the available package is available
available("available")

# You can disable opening of browser windows with browse = FALSE
available("survival", browse = FALSE)

# Or by setting a global option
options(available.browse = FALSE)
available("survival")

# Test if a name is available in a non-default CRAN repository by setting
# the 'repos' argument
available_on_cran("semaforos")
available_on_bioc

```r
available_on_cran("semaforos", repos = "https://bisaloo.r-universe.dev")
## End(Not run)
```

---

**available_on_bioc**  
*See if a name is available on CRAN*

## Description

See if a name is available on CRAN

## Usage

```r
available_on_bioc(name, repos = NULL, ...)
available_on_cran(name, repos = default_cran_repos, ...)
```

### Arguments

- `name`  
  Name of package to search

- `repos`  
  character vector, the base URL(s) of the repositories to use.

- `...`  
  Additional arguments passed to `utils::available.packages()`.

## Examples

```r
available_on_cran("semaforos")
```

# Test if this name is available in a non-default CRAN repository
## Not run:
```
available_on_cran("semaforos", repos = "https://bisaloo.r-universe.dev")
## End(Not run)
```

---

**available_on_github**  
*See if a name is available on github*

## Description

See if a name is available on github

## Usage

```r
available_on_github(name)
```

### Arguments

- `name`  
  Name of package to search
### create

*Check a new package name and possibly create it*

**Description**

Check a new package name and possibly create it

**Usage**

```r
create(name, ...)  
```

**Arguments**

- `name`: Name of package to search
- `...`: Additional arguments passed to `usethis::create_package()`.

### get_bad_words

*Check for bad words in name*

**Description**

Check for bad words in name

**Usage**

```r
get_bad_words(name)  
```

**Arguments**

- `name`: Name of package to search

**See Also**

See [https://github.com/web-mech/badwords](https://github.com/web-mech/badwords)
**get_sentiment**

*Get sentiment of name*

**Description**

Get sentiment of name

**Usage**

```python
get_sentiment(name)
```

**Arguments**

- `name`: Name of package to search

---

**get_wikipedia**

*Open wikipedia page and abbreviations.com page*

**Description**

Open wikipedia page and abbreviations.com page

**Usage**

```python
get_wikipedia(name)
```

```python
get_wiktionary(name)
```

**Arguments**

- `name`: Name of package to search

---

**pkg_name_dist**

*Find five closest package names in terms of string distance*

**Description**

Find five closest package names in terms of string distance

**Usage**

```python
pkg_name_dist(name, pkgs)
```

**Arguments**

- `name`: name of package
- `pkgs`: packages to compare with
suggest

Suggest a package name based on a development package title or description

**Description**

If the package you are using already has a title, simply pass the path to the package root in `path`. Otherwise use `title` to specify a potential title.

**Usage**

```r
suggest(path = ".", field = c("Title", "Description"), text = NULL)
```

**Arguments**

- `path` : Path to a existing package to extract the title from.
- `field` : one of "Title" or "Description"
- `text` : text string to search.

**Examples**

```r
# Not run:
# Default will use the title from the current path.
suggest()

# Can also suggest based on the description
suggest(field = "Description")

# Or by explicitly using the text argument
suggest(text = "A Package for Displaying Visual Scenes as They May Appear to an Animal with Lower Acuity")
```

valid_package_name

Is a package name valid

**Description**

Is a package name valid

**Usage**

```r
valid_package_name(name)
```

**Arguments**

- `name` : Name of package to search
Index

available, 2
available_on_bioc, 3
available_on_cran(available_on_bioc), 3
available_on_github, 3

create, 4

get_bad_words, 4
get_sentiment, 5
get_wikipedia, 5
get_wiktionary(get_wikipedia), 5

pkg_name_dist, 5

Startup, 2
suggest, 6

usethis::create_package(), 4
utils::available.packages(), 2, 3

valid_package_name, 6