Package 'devtoolbox'

September 7, 2021

Type Package

Title Tools for the R Developer

Version 0.1.0

Description Reporting tools for the R developer to evaluate their packages in terms of complexity, usage, and performance. Developers can generate an HTML report that displays CRAN downloads, number of open GitHub issues and pull requests, package dependencies, and so on, with each component of the report available as independent functions.

URL https://github.com/martinctc/devtoolbox/

BugReports https://github.com/martinctc/devtoolbox/issues/

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.1.2)

Imports dplyr, stats, utils, tidyr, tidyselect (>= 1.0.0), magrittr, purrr, reshape2, ggplot2, markdown, rmarkdown, stringr, DT, gh, htmltools, jsonlite, lubridate, flexdashboard, httr, ggiraph

RoxygenNote 7.1.1

Suggests knitr, glue

Language en-US

NeedsCompilation no

Author Martin Chan [aut, cre]

Maintainer Martin Chan <martinchan53@gmail.com>

Repository CRAN

Date/Publication 2021-09-07 09:20:06 UTC

R topics documented:

`create_pkg_report` *Generate a package report in HTML*

Description

Create a report in HTML

Usage

```
create_pkg_report(
  pkgname,
  from = NULL,
  to = NULL,
  path = paste0(pkgname, " - summary report", from, "_", to),
  gh = NULL
)
```

Arguments

`pkgname`	String containing the name of the package.
`from`	String in the format of `YYYY-MM-DD` specifying the start date of the reporting period. If set to `NULL`, the first day of the month of the system date will be used.
`to`	String in the format of `YYYY-MM-DD` specifying the end date of the reporting period. If set to `NULL`, the last day of the month of the system date will be used.
`path`	Pass the file path and the desired file name, *excluding the file extension*. For example, `"my package report"`.
`gh`	String in the format of `owner/repo` to specify the GitHub repository that holds the package. Defaults to `NULL` by default, where GitHub statistics will be omitted.

Value

Opens and saves a static HTML file in the active working directory with the file name specified in `path`.

extract_argument *Extract the arguments of a function for a loaded package*

Description

Returns a character vector containing the arguments of a function for a loaded package. Arguments are separated by semi-colons.

Usage

```
extract_argument(fnc_name, package_name)
```

Arguments

fnc_name	String containing the name of the function.
package_name	String containing the name of the loaded package.

Value

A character vector of length 1 containing the arguments of a function for a loaded package. Multiple arguments are separated by a semi-colon.

Examples

```
extract_argument(
  fnc_name = "extract_argument",
  package_name = "devtoolbox")
```

extract_desc *Extract the description of a function for a loaded package*

Description

Returns a character vector containing the description of a function for a loaded package.

Usage

```
extract_desc(fnc_name, package_name)
```

Arguments

fnc_name	String containing the name of the function.
package_name	String containing the name of the loaded package.

Value

A character vector containing the description of a function for a loaded package.

Examples

```
extract_desc(
  fnc_name = "extract_desc",
  package_name = "devtoolbox")
```

`extract_family` *Extract the family of a function for a loaded package*

Description

Returns a character vector containing the family of a function for a loaded package.

Usage

```
extract_family(fnc_name, package_name)
```

Arguments

`fnc_name`	String containing the name of the function.
`package_name`	String containing the name of the loaded package.

Value

A character vector containing the family of a function for a loaded package.

Examples

```
extract_family(
  fnc_name = "extract_family",
  package_name = "devtoolbox")
```

extract_title *Extract the title of a function for a loaded package*

Description

Returns a character vector containing the title of a function for a loaded package.

Usage

```
extract_title(fnc_name, package_name)
```

Arguments

fnc_name	String containing the name of the function.
package_name	String containing the name of the loaded package.

Value

A character vector containing the title of a function for a loaded package.

Examples

```
extract_title(
  fnc_name = "extract_title",
  package_name = "devtoolbox")
```

generate_summary *Run a summary of functions and the associated description for a loaded package*

Description

Return a data frame summarising functions, family, title, description, and arguments of a loaded package.

Usage

```
generate_summary(package_name)
```

Arguments

package_name	String providing the name of the loaded package, e.g. surveytoolbox

Value

A data frame summarising functions, family, title, description, and arguments of a loaded package.

Examples

```
library(devtoolbox)
generate_summary(package_name = "devtoolbox")
```

`get_gh_issues` *Get issues information from GitHub for the given time period.*

Description

Get information on issues from the specified GitHub repository. This is a wrapper around the 'gh' package. Original code taken from https://github.com/jennybc/analyze-github-stuff-with-r/.

Usage

```
get_gh_issues(owner, repo, start_date = NULL, end_date = NULL)
```

Arguments

`owner`	String specifying the owner of the repository.
`repo`	String specifying the name of the repo.
`start_date`	String specifying start date of reporting period to filter by, in the format `YYYY-MM-DD`. `NULL` by default, where no filters will be applied.
`end_date`	String specifying end date of reporting period to filter by, in the format `YYYY-MM-DD`. `NULL` by default, where no filters will be applied.

Value

A data frame containing details of GitHub issues identified in the specified date range.

Examples

```
get_gh_issues(owner = "martinctc", repo = "rwa")
```

get_gh_pr *Get pull request information from GitHub for the given time period.*

Description

Get information on pull requests from the specified GitHub repository. This is a wrapper around the 'gh' package. Original code taken from https://github.com/jennybc/analyze-github-stuff-with-r/.

Usage

```
get_gh_pr(owner, repo, start_date = NULL, end_date = NULL)
```

Arguments

owner	String specifying the owner of the repository.
repo	String specifying the name of the repo.
start_date	String specifying start date of reporting period to filter by, in the format YYYY-MM-DD. NULL by default, where no filters will be applied.
end_date	String specifying end date of reporting period to filter by, in the format YYYY-MM-DD. NULL by default, where no filters will be applied.

Value

A data frame containing details of GitHub pull requests identified in the specified date range.

Examples

```
get_gh_pr(owner = "martinctc", repo = "rwa")
```

recursive_dependencies *Compute recursive dependencies of packages*

Description

Compute recursive package dependencies of packages. Credits to 'pkgnet' for the original function.

Usage

```
recursive_dependencies(package, db, seen_packages = NULL)
```

Arguments

package	String specifying name of original package.
db	Parameter to pass to 'tools::package_dependencies'
seen_packages	String containing names of packages to exclude

Value

A character vector containing all the recursive package dependencies of the specified package.

---

run_rmd *Run RMarkdown Report based on an existing RMarkdown file*

Description

This is a support function that accepts parameters and creates a HTML document based on an RMarkdown template. This function is taken from the `generate_report2()` function from the 'wpa' package.

Usage

```
run_rmd(
 output_format = rmarkdown::html_document(toc = TRUE, toc_depth = 6, theme = "cosmo"),
  output_file = "report.html",
  output_dir = getwd(),
  report_title = "Report",
  rmd_dir = system.file("rmd_template/minimal.rmd", package = "devtoolbox"),
  ...
)
```

Arguments

output_format	output format in `rmarkdown::render()`. Default is `rmarkdown::html_document(toc = TRUE,toc_depth = 6,theme = "cosmo")`.
output_file	output file name in `rmarkdown::render()`. Default is `"report.html"`.
output_dir	output directory for report in `rmarkdown::render()`. Default is user's current directory.
report_title	report title. Default is `"Report"`.
rmd_dir	string specifying the path to the directory containing the RMarkdown template files. This uses the internal `minimal` template by default.
...	other arguments to be passed to `params`.

Value

Opens and saves a static HTML report in the active directory, using the RMarkdown template as specified in the argument `rmd_dir`.

Running the report

You can run the minimal report and pass arguments directly to `run_rmd`:

```
run_rmd(pkgname = "devtoolbox")
```

Note

The implementation of this function was inspired by the 'DataExplorer' package by boxuancui, with credits due to the original author.

Index