Package 'DependencyReviewer'

January 9, 2023

Type Package

Title Tool Suite to Investigate R package dependencies

Version 1.0.0

Language en-US

Description Helps investigating other packages during code review by looking at their dependencies.

License Apache License (>= 2)

Encoding UTF-8

LazyData true

Suggests rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports rlang, desc, DT, cli, dplyr, ggplot2, ggraph, glue, here, knitr, lintr, pak, readr, shiny, shinyAce, stringr, tidygraph, tidyverse, magrittr, GGally

RoxygenNote 7.2.2

Depends R (>= 4.0)

VignetteBuilder knitr

R topics documented:

`checkDependencies` *checkDependencies*

Description

Check package dependencies

Usage

```
checkDependencies(packageName = NULL, dependencyType = c("Imports", "Depends"))
```

Arguments

`packageName`	Name of package to profile. If NULL current package
`dependencyType`	Imports, depends, and/ or suggests

Value

Returns value NULL

Examples

```
# Run only in interactive session
if (interactive()) {
  checkDependencies(packageName = "DependencyReviewer")

  checkDependencies(packageName = "DependencyReviewer", c("Suggests"))
}
```

darwinLintFile *darwinLintFile*

Description

Lint a given file.

Usage

```
darwinLintFile(fileName)
```

Arguments

fileName	Path to file to lint

Value

list of lint objects.

Examples

```
darwinLintFile(
  fileName = system.file(package = "DependencyReviewer", "testScript.R"))
```

darwinLintPackage *darwinLintPackage*

Description

Darwin lintr object, using default lintr object with camelCase

Usage

```
darwinLintPackage()
```

Value

List of lint objects.

Examples

```
darwinLintPackage()
```

`darwinLintScore` *darwinLintScore*

Description

Function that scores the lintr output as a percentage per message type (style, warning, error). Lintr messages / lines assessed * 100

Usage

```
darwinLintScore(lintFunction, ...)
```

Arguments

`lintFunction`	Lint function to use
`...`	Other parameters a Lint function might need (i.e. file name)

Value

A tibble of percentage scores per type of Lint message.

Examples

```
# With darwin file lintr
darwinLintScore(
  lintFunction = darwinLintFile,
  system.file(package = "DependencyReviewer", "testScript.R"))

# With standard package lintr
darwinLintScore(
  lintFunction = lintr::lint_package,
  system.file(package = "DependencyReviewer"))
```

`funsUsedInFile` *funsUsedInFile*

Description

Support function

Usage

```
funsUsedInFile(files, verbose = FALSE, in_package = TRUE)
```

Arguments

`files`	Files to get functions from
`verbose`	Verbosity
`in_package`	default: TRUE

Value

table

funsUsedInLine *funsUsedInLine*

Description

Support function for funsUsedInFile.

Usage

```
funsUsedInLine(file_txt, file_name, i, verbose = FALSE)
```

Arguments

file_txt	file to use
file_name	name of file
i	line
verbose	Prints message when no function found

Value

data.frame of 3 colums: Package (pkg); Function (fun); Line in script (line)

---

getDefaultPermittedPackages *getDefaultPermittedpackages*

Description

Gets permitted packages

Usage

```
getDefaultPermittedPackages()
```

Value

tibble of two columns (package, version) with all 'allowed' packages.

Examples

```
# Run only in interactive session
if (interactive()) {
  getDefaultPermittedPackages()
}
```

getDiffVersions *getDiffVersions*

Description

Helper function

Usage

```
getDiffVersions(dependencies, permittedPackages)
```

Arguments

dependencies	Dependencies
permittedPackages	permittedPackages

Value

versions of permitted packages

getGraphData *getGraphData*

Description

getGraphData

Usage

```
getGraphData(
  path = here::here(),
  excluded_packages = c(""),
  package_types = c("imports", "depends")
)
```

Arguments

path	path
excluded_packages	Packages to exclude
package_types	Types of packages to be included in the result. Default: c("imports", "depends") Availible types are: "imports", "depends", "suggests", "enhances", "linkingto"

Value

net_data graph data

Examples

```
# Only run in interactive session
if(interactive()) {
  graphData <- getGraphData()
}
```

`getNotPermitted` *getNotPermitted*

Description

Helper function

Usage

```
getNotPermitted(dependencies, permittedPackages)
```

Arguments

`dependencies`	Dependencies
`permittedPackages`	Packages that are permitted as character vector

Value

Returns vector of not permitted packages

`messagePackageVersion` *messagePackageVersion*

Description

Helper message function

Usage

```
messagePackageVersion(i, diffVersions)
```

Arguments

`i`	iterator
`diffVersions`	different versions

`messagePermission` *messagePermission*

Description

Helper message function

Usage

```
messagePermission(i, not_permitted)
```

Arguments

`i`	iterator
`not_permitted`	Not permitted

`runShiny` *runShiny*

Description

Runs a Shiny app for dependency investigation.

Usage

```
runShiny()
```

Value

An object that represents the app.

Examples

```
# Run only in interactive session
if (interactive()) {
  runShiny()
}
```

summariseFunctionUse *summariseFunctionUse*

Description

Summarise functions used in R package

Usage

```
summariseFunctionUse(
  r_files = list.files(here::here("R")),
  verbose = FALSE,
  in_package = TRUE
)
```

Arguments

r_files	r_files
verbose	Default: FALSE; prints message to console which file is currently being worked on.
in_package	Default: TRUE; Indicate if the function is called within a package project or not. TRUE: expects a file name "myFile.R", may be a vector of multiple. FALSE: expects a file path "./my/file/path/myFile.R", my be a vector of multiple

Value

tibble

Examples

```
summariseFunctionUse(
  r_files = system.file(package = "DependencyReviewer", "testScript.R"),
  in_package = FALSE)

# Only in an interactive session
if (interactive()) {
  summariseFunctionUse()
}
```

Index