

Package ‘DependencyReviewer’

January 12, 2023

Type Package

Title Tool Suite to Investigate Other Packages

Version 1.1.0

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 testthat (>= 3.0.0),

shiny,
shinyAce,
ggplot2,
ggraph,
DT,
GGally,
here,
igraph,
withr,
network,
sna,
intergraph

Imports rlang,

desc,
cli,
dplyr,
knitr,
lintr,
pak,
readr,
stringr,
tidygraph,
magrittr

RoxygenNote 7.2.3

Depends R (>= 4.0)

VignetteBuilder knitr

R topics documented:

checkDependencies	2
darwinLintFile	3
darwinLintPackage	3
darwinLintScore	4
funsUsedInFile	4
funsUsedInLine	5
getDefaultPermittedPackages	5
getDiffVersions	6
getGraphData	6
getNotPermitted	7
messagePackageVersion	7
messagePermission	8
runShiny	8
summariseFunctionUse	9
Index	10

checkDependencies	<i>checkDependencies</i>
-------------------	--------------------------

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	<i>darwinLintFile</i>
----------------	-----------------------

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	<i>darwinLintPackage</i>
-------------------	--------------------------

Description

Darwin lintr object, using default lintr object with camelCase

Usage

```
darwinLintPackage(path = ".")
```

Arguments

path	Default: ".", Path to package
------	-------------------------------

Value

List of lint objects.

Examples

```
darwinLintPackage()
```

darwinLintScore	<i>darwinLintScore</i>
-----------------	------------------------

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	<i>funsUsedInFile</i>
----------------	-----------------------

Description

Support function

Usage

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

Arguments

files	Files to get functions from
verbose	Verbosity

Value

table

funcsUsedInLine	<i>funcsUsedInLine</i>
-----------------	------------------------

Description

Support function for funcsUsedInFile.

Usage

```
funcsUsedInLine(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 columns: Package (pkg); Function (fun); Line in script (line)

getDefaultPermittedPackages	<i>getDefaultPermittedPackages</i>
-----------------------------	------------------------------------

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	<i>getDiffVersions</i>
-----------------	------------------------

Description

Helper function

Usage

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

Arguments

dependencies	Dependencies
permittedPackages	permittedPackages

Value

versions of permitted packages

getGraphData	<i>getGraphData</i>
--------------	---------------------

Description

getGraphData

Usage

```
getGraphData(
  path = "./",
  excluded_packages = c(""),
  package_types = c("imports", "depends")
)
```

Arguments

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

Value

net_data graph data

Examples

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

getNotPermitted	<i>getNotPermitted</i>
-----------------	------------------------

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	<i>messagePackageVersion</i>
-----------------------	------------------------------

Description

Helper message function

Usage

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

Arguments

i	iterator
diffVersions	different versions

messagePermission	<i>messagePermission</i>
-------------------	--------------------------

Description

Helper message function

Usage

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

Arguments

i	iterator
not_permitted	Not permitted

runShiny	<i>runShiny</i>
----------	-----------------

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, verbose = FALSE)
```

Arguments

<code>r_files</code>	Complete path(s) to files to be investigated
<code>verbose</code>	Default: FALSE; prints message to console which file is currently being worked on.

Value

tibble

Examples

```
summariseFunctionUse(  
  r_files = system.file(package = "DependencyReviewer", "testScript.R"))  
  
# Only in an interactive session  
if (interactive()) {  
  summariseFunctionUse(list.files(here::here("R"), full.names = TRUE))  
}
```

Index

checkDependencies, [2](#)

darwinLintFile, [3](#)
darwinLintPackage, [3](#)
darwinLintScore, [4](#)

funcsUsedInFile, [4](#)
funcsUsedInLine, [5](#)

getDefaultPermittedPackages, [5](#)
getDiffVersions, [6](#)
getGraphData, [6](#)
getNotPermitted, [7](#)

messagePackageVersion, [7](#)
messagePermission, [8](#)

runShiny, [8](#)

summariseFunctionUse, [9](#)