

Package ‘functiondepends’

November 3, 2020

Type Package

Title Find Functions and their Dependencies

Version 0.2.1

Description Find functions in an unstructured directory and explore their dependencies.
Sourcing of R source files is performed without side-effects: from R scripts that have executable code and function definitions only functions are sourced.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests testthat, covr, knitr, rmarkdown, ggplot2, igraph

Imports dplyr (>= 1.0.1), tidyr (>= 1.1.1), stringr (>= 1.4.0), purrr (>= 0.3.4), magrittr (>= 1.5), tibble (>= 3.0.3), tidyselect (>= 1.1.0)

RoxygenNote 7.1.1

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Author Jakub Sobolewski [aut, cre]

Maintainer Jakub Sobolewski <jakupsob@gmail.com>

Repository CRAN

Date/Publication 2020-11-03 12:30:02 UTC

R topics documented:

find_dependencies	2
find_functions	2

Index	4
--------------	----------

find_dependencies	<i>Find dependencies</i>
-------------------	--------------------------

Description

This function finds function calls inside a function with given name. Be aware that any variable that has a name that overwrites a function name will be recognised as a function call.

Usage

```
find_dependencies(
  function_name,
  envir = .GlobalEnv,
  in_envir = TRUE,
  add_info = FALSE
)
```

Arguments

function_name	Character, name of function
envir	Environment in which to search for function. Default is .GlobalEnv
in_envir	Logical, whether to return only functions that are loaded into envir
add_info	Logical, whether to add list column with line numbers of given function call in function body and a list column with context of said calls. Default is FALSE.

Value

A tibble with columns: - Source: character, name of function called inside 'Target' - SourceRep: integer, number of times 'Source' is called - SourceNamespace: character, name of namespace from which the function comes, if a function is defined in multiple namespaces then it is a vector. If function is user defined 'Namespace' is NA. - SourcePosition: optional, integer list with positions of 'Source' calls in body - SourceContext: optional, character list with lines of code with calls of 'Source' - Target: character, name of inspected function - TargetInDegree: integer, number of function calls inside of function body

find_functions	<i>Functions in path</i>
----------------	--------------------------

Description

Parses files in given path. It searches for functions and loads them. Is safe for use with scripts as it doesn't source the whole file, just functions. There are no side-effects to sourcing .R files.

Usage

```
find_functions(path, envir = .GlobalEnv, recursive = TRUE)
```

Arguments

<code>path</code>	Character, path to folder
<code>envir</code>	Environment to source loaded functions into
<code>recursive</code>	Logical, whether to search files recursively

Value

A tibble with character columns indicating path to source files and names of functions defined in them. Path elements are split into columns with 'Level' prefix, name of source file is in 'Source' column, name of function is in 'Function' column.

Examples

```
path <- file.path(tempdir(), "find_functions_example")
dir.create(path, showWarnings = FALSE)
code <- "
add <- function(x, y) {
  x + y
}
add_one = function(x) {
  add(x, 1)
}
assign('add_two', function(x) {
  add(x, 2)
})
"
write(code, file.path(path, "code.R"))
find_functions(path)
```

Index

`find_dependencies`, [2](#)
`find_functions`, [2](#)