Package 'sinew'

October 28, 2017

Type Package

Title Create 'roxygen2' Skeleton with Information from Function Script

Version 0.3.5.1

Date 2017-10-28

Author c(person(``Jonathan'', ``Sidi'', email = ``yonis@metrumrg.com'', role = c(``aut'', ``cre'')), person(``Anton'',``Grishin'',role = ``ctb''), person(``Lorenzo'',``Busetto'',role = ``ctb''))

Maintainer Jonathan Sidi <yonis@metrumrg.com>

Description Create 'roxygen2' skeleton populated with information scraped from the within the function script. Also creates field entries for imports in the 'DESCRIPTION' and import in the 'NAMESPACE' files. Can be run from the R console or through the 'RStudio' 'addin' menu.

Depends R (>= 2.3.0)

Imports rstudioapi,utils,shiny,miniUI,tools,sos,stringi,yaml

Suggests roxygen2

License GPL-2 | GPL-3

URL https://github.com/metrumresearchgroup/sinew

BugReports https://github.com/metrumresearchgroup/sinew/issues

LazyData true

NeedsCompilation no

RoxygenNote 6.0.1

R topics documented:

1

Index   **14**

---

create_yml   *Create _sinewconfig.yml*

Description

Create _sinewconfig.yml file in project root directory

Usage

```
create_yml()
```

Value

nothing

Author(s)

Jonathan Sidi

Examples

```
## Not run:
create_yml()

## End(Not run)
```

---

interOxyAddIn   *Interactive add-in*

Description

Launches an interactive addin for insertion of roxygen2 comments in files. Allows selection of extra parameters for makeOxygen

Usage

```
interOxyAddIn()
```

Details

Open an .R file in Rstudio's source editor. Launch the add-in via Addins -> interactiveOxygen or interOxyAddIn() in the console. Add-in opens in the viewer panel. Select function's/dataset's name in the source editor. If objects cannot be found, the addin prompts to source the file. Choose parameters for makeOxygen. Click Insert. Select next object's name. Rinse.Repeat. Click Quit when done with the file.

Value

Nothing. Inserts roxygen2 comments in a file opened in the source editor.

Author(s)

Anton Grishin, Jonathan Sidi

See Also

```
View(sinew:::oxygenAddin)
```

Examples

```
if(interactive()) interOxyAddIn()
```

---

ls_param	*Return roxygen2 parameter calls from parameter dictionary*

Description

Return roxygen2 parameter calls from the intersection of the parameters listed in the package dictionary and the formals of a function

Usage

```
ls_param(obj, dictionary = "man-roxygen/Dictionary-1.R", print = TRUE)
```

Arguments

obj	function or name of function
dictionary	character, path_to_dictionary, Default: 'roxygen-man/Dictionary-1.R'
print	boolean print output to console, Default: TRUE

Value

character vector

Examples

```
repo='https://raw.githubusercontent.com/metrumresearchgroup/sinew/master/'
dict_loc=file.path(repo,'man-roxygen/Dictionary-1.R')
ls_param(sinew::makeOxygen,dictionary=dict_loc)
```

---

makeDictionary	*Parse package R files to create dictionary of unique parameter definitions*

Description

Given list of R files function returns roxygen2 template consisting of intersecting parameter definitions

Usage

```
makeDictionary(path, save_path = FALSE)
```

Arguments

path	character or character vector of paths to files to parse
save_path	boolean that allows for function to write template to man-roxygen subdirectory, Default: FALSE

Value

character/character vector of intersecting parameters

Examples

```
makeDictionary('R')
```

---

makeImport	*Scrape R script to create namespace calls for R documentation*

Description

Scrape r script to create namespace calls for roxygen2, namespace or description files

Usage

```
makeImport(script, cut = NULL, print = TRUE, format = "oxygen",
  desc_loc = NULL)
```

Arguments

script	character, connection to pass to readLines, can be file path, directory path, url path
cut	integer, number of functions to write as importFrom until switches to import, Default: NULL
print	boolean, print output to console, Default: TRUE
format	character, the output format must be in c('oxygen','description'), Default: 'oxygen'
desc_loc	character, path to DESCRIPTION file, if not NULL then the Imports fields in the DESCRIPTION file, Default: NULL

Examples

```
makeImport('R',format = 'oxygen')
makeImport('R',format = 'description')
```

---

makeOxyFile   *Inserts roxygen2 skeletons in file(s).*

Description

Applies makeOxygen function to all functions/dataframes in supplied file(s)

Usage

```
makeOxyFile(input = NULL, overwrite = FALSE, verbose = TRUE, ...)
```

Arguments

input	character, vector of path(s) to one or more .R files, a path to directory containing .R files, Default: NULL
overwrite	boolean, If TRUE overwrites file(s), FALSE writes "Oxy"- prefixed files in the same directory, Default: FALSE
verbose	boolean, If TRUE will print output to console and open edited files in the editor viewer, Defulat: TRUE
...	additional parameters passed to makeOxygen

Details

If an object cannot be found it will be sourced into a temporary environment. If the file already contains roxygen2 comments they will be deleted to avoid duplication. Some functions may require attaching additional packages. For instance, if functions were defined with purr's compose or partial functions, omission of purr:: in definitions will require library(purrr) before proceding with makeOxyFile.

Value

Nothing. Writes files with roxygen2 comments as a side effect

Author(s)

Anton Grishin

See Also

[makeOxygen](#)

Examples

```
## Not run:
if(interactive()){
 makeOxyFile() # default args, opens system file selection dialogue
 }
 makeOxyFile("./myRfunctions/utils.R") # on one R file
 makeOxyFile("./myRfunctions/") # on all R files in directory

 # adds more fields to defaults, passes "cut" to makeImport

 sinew_opts$append(list(add_fields=c("concept", "describeIn")))
 makeOxyFile("./myRfunctions/utils.R", cut = 5)

## End(Not run)
```

---

makeOxygen	*Creates skeleton roxygen2 with information from within function script*

Description

Creates roxygen2 skeleton including title, description, import and other fields

Usage

```
makeOxygen(obj, add_default = TRUE,
  add_fields = sinew_opts$get("add_fields"), use_dictionary = NULL,
  print = TRUE, ...)
```

Arguments

obj	function or name of function
add_default	boolean to add defaults values to the end of the PARAM fields, Default: TRUE
add_fields	character vector to add additional roxygen2 fields, Default: c("details","examples","seealso","rdname
use_dictionary	character, path_to_dictionary, Default: NULL
print	boolean print output to console, Default: TRUE
...	arguments to be passed to makeImport

Details

add_fields can include any slot except for the defaults (title,description,param,return). The order in add_fields determines the order of printout. The roxygen2 fields to add are list below, for more information go to Generating Rd files

Field	Skeleton
author	AUTHOR [AUTHOR_2]
backref	src/filename.cpp
concept	CONCEPT_TERM_1 [CONCEPT_TERM_2]
describeIn	FUNCTION_NAME DESCRIPTION
details	DETAILS
example	path/relative/to/packge/root

export	
family	FAMILY_TITLE
field	FIELD_IN_S4_RefClass DESCRIPTION
format	DATA_STRUCTURE
importClassesFrom	PKG CLASS_a [CLASS_b]
importMethodsFrom	PKG METHOD_a [METHOD_b]
include	FILENAME.R [FILENAME_b.R]
inherit	[PKG::]SOURCE_FUNCTION [FIELD_a FIELD_b]
inheritDotParams	[PKG::]SOURCE_FUNCTION
inheritSection	[PKG::]SOURCE_FUNCTION [SECTION_a SECTION_b]
keywords	KEYWORD_TERM
name	NAME
rdname	FUNCTION_NAME
references	BIB_CITATION
section	SECTION_NAME
source	\url{http://somewhere.important.com/}
slot	SLOTNAME DESCRIPTION
template	FILENAME
templateVar	NAME VALUE
useDynLib	PKG [routine_a routine_b]

Examples

```
makeOxygen(stats::lm)
```

---

moga	*Make Oxygen Great Again*

Description

Update/append an R file that has roxygen2 headers already with updated information

Usage

```
moga(path, ..., force.fields = NULL, dry.run = TRUE, overwrite = FALSE)
```

Arguments

`path`	character path to R file
`...`	arguments to be passed to new makeOxygen
`force.fields`	character, vector a field names that are in current header that are to be updated Default: NULL
`dry.run`	boolean, write lines to console the output, Default: TRUE
`overwrite`	boolean, overwrite contents of input file, Default: FALSE

Details

Cross references fields already in the roxygen2 header and adds any new ones from the updated call. To force a change to a field add field name to force.fields.

Value

character

Examples

```
moga('https://raw.githubusercontent.com/metrumresearchgroup/ggedit/master/R/aesColour.R')
```

---

pretty_namespace   *Append namespace to functions in script*

Description

Autoappend namespace to functions in script by searchpath order

Usage

```
pretty_namespace(con = NULL, text = NULL, overwrite = FALSE,
  sos = FALSE)
```

Arguments

con	character, path to file or directory that contains script, Default: NULL
text	character, vector that contains script, Default: NULL
overwrite	boolean, overwrite original file, Default: FALSE
sos	boolean, apply sos search for uninstalled libraries, Default: FALSE

Details

searches for functions in the loadedNamespace, help.search and then findFn

Value

character

Author(s)

Jonathan Sidi

See Also

findFn help.search

Examples

```
txt <- '#some comment
yy <- function(a=4){
  head(runif(10),a)
  # a comment
}

zz <- function(v=10,a=8){
  head(runif(v),a)
}'

pretty_namespace(text=txt)
```

---

rmOxygen *Remove roxygen2 Comments From an .R File*

Description

Strips .R files of roxygen2 style comments (#')

Usage

```
rmOxygen(.file)
```

Arguments

.file,   path to an .R file, character vector of length 1

Value

Nothing. Overwrites files as a side effect

Author(s)

Anton Grishin

Examples

```
## Not run:
rmOxygen("./myRfunctions/function1.R")

## End(Not run)
```

---

sinew_opts *Default and current sinew options*

Description

Options for functions in the sinew package. When running R code, the object `sinew_opts` (default options) is not modified by chunk headers (local chunk options are merged with default options), whereas `sinew_opts_current` (current options) changes with different chunk headers and it always reflects the options for the current chunk.

Usage

```
sinew_opts

sinew_opts_current
```

Format

An object of class `list` of length 5.

Details

Normally we set up the global options once in the first code chunk in a document using `sinew_opts$set()`, so that all *latter* chunks will use these options. Note the global options set in one chunk will not affect the options in this chunk itself, and that is why we often need to set global options in a separate chunk.

Below is a list of default chunk options, retrieved via `sinew_opts$get()`:

```
List of 28
 $ add_fields     : chr [1:6] "details" "examples" "seealso" "rdname" ...
 $ author         : chr "Jonathan Sidi"
 $ backref        : chr "src/filename.cpp"
 $ concept        : chr "CONCEPT_TERM_1 [CONCEPT_TERM_2]"
 $ describeIn     : chr "FUNCTION_NAME DESCRIPTION"
 $ details        : chr "DETAILS"
 $ example        : chr "path_to_file/relative/to/packge/root"
 $ examples       : chr "\n#' \\dontrun{\n#' if(interactive()){\n#'  #EXAMPLE1\n#'  }\n#' }"
 $ export         : chr ""
 $ family         : chr "FAMILY_TITLE"
 $ field          : chr "FIELD_IN_S4_RefClass DESCRIPTION"
 $ format         : chr "DATA_STRUCTURE"
 $ importClassesFrom: chr "PKG CLASS_a [CLASS_b]"
 $ importMethodsFrom: chr "PKG METHOD_a [METHOD_b]"
 $ include        : chr "FILENAME.R [FILENAME_b.R]"
 $ inherit        : chr "[PKG::]SOURCE_FUNCTION [FIELD_a FIELD_b]"
 $ inheritDotParams : chr "[PKG::]SOURCE_FUNCTION"
 $ inheritSection   : chr "[PKG::]SOURCE_FUNCTION [SECTION_a SECTION_b]"
 $ keywords       : chr "KEYWORD_TERM"
 $ name           : chr "NAME"
 $ rdname         : chr "FUNCTION_NAME"
 $ references     : chr "BIB_CITATION"
 $ section        : chr "SECTION_NAME"
```

```
$ source       : chr "\\url{http://somewhere.important.com/}"
$ slot         : chr "SLOTNAME DESCRIPTION"
$ template     : chr "FILENAME"
$ templateVar  : chr "NAME VALUE"
$ useDynLib    : chr "PKG [ROUTINE_a ROUTINE_b]"
```

Note

sinew_opts_current is read-only in the sense that it does nothing if you call sinew_opts_current$set(); you can only query the options via sinew_opts_current$get().

Examples

```
sinew_opts$get('add_fields')
```

---

tabular	*Tabular for roxygen2*

Description

Convert data.frame to roxygen2 tabular format

Usage

```
tabular(df, header = TRUE, ...)
```

Arguments

df	data.frame to convert to table
header	boolean to control if header is created from names(df), Default: TRUE
...	arguments to pass to format

Value

character

Source

roxygen2 formatting

See Also

format

Examples

```
cat(tabular(mtcars[1:5, 1:5]))
cat(tabular(mtcars[1:5, 1:5],header=FALSE))
```

---

untangle   *Split an R script by functions*

Description

Split a R script with multiple functions into multiple single function R files.

Usage

```
untangle(text = NULL, file = "", dir.out = NULL, keep.body = TRUE)
```

Arguments

text	character, vector of R commands, Default: NULL
file	character, path to R file, Default: ''
dir.out	character, path to save new R files, Default: NULL
keep.body	boolean, if TRUE all non-funcitons will be saved to body.R, Default: TRUE

Details

body.R is written to the working directory and not dir.out .

Value

list of seperate functions

Author(s)

Jonathan Sidi

Examples

```
## Not run:
txt <- "#some comment
#' @import utils
yy <- function(a=4){
 utils::head(runif(10),a)
 # a comment
}

v <- 20

#another comment
#' @import utils
zz <- function(v=10,a=3){
 utils::head(runif(v),pmin(a,v))
}

zz(v)

"
untangle(txt,dir.out = 'test')
```

```
## End(Not run)
```

Index