

Package ‘namer’

March 26, 2019

Title Names Your 'R Markdown' Chunks

Version 0.1.4

Description It names the 'R Markdown' chunks of files based on the filename.

License GPL-3

Encoding UTF-8

LazyData true

Imports fs, glue, magrittr, purrr, tibble, dplyr, rstudioapi

RoxygenNote 6.1.1

Suggests testthat, covr, rmarkdown, knitr, UNF

VignetteBuilder knitr

URL <https://github.com/lockedata/namer>

BugReports <https://github.com/lockedata/namer/issues>

NeedsCompilation no

Author Steph Locke [aut, cre],
Maëlle Salmon [aut] (<<https://orcid.org/0000-0002-2815-0399>>),
Ellis Valentiner [ctb],
Martin Hadley [ctb] (<<https://orcid.org/0000-0002-3039-6849>>),
Locke Data [fnd] (<https://itsalocke.com>)

Maintainer Steph Locke <steph@itsalocke.com>

Repository CRAN

Date/Publication 2019-03-26 14:10:02 UTC

R topics documented:

name_chunks	2
name_dir_chunks	2
unname_all_chunks	3

Index	5
--------------	----------

name_chunks	<i>Name chunks in a single file</i>
-------------	-------------------------------------

Description

Name unnamed chunks in a single file using the filename with extension stripped as basis.

Usage

```
name_chunks(path)
```

Arguments

path	Path to file
------	--------------

Details

When using namer, please check the edits before pushing them to your code base. Such automatic chunk labelling is best paired with version control.

Examples

```
temp_file_path <- file.path(tempdir(), "test.Rmd")
file.copy(system.file("examples", "example1.Rmd", package = "namer"),
          temp_file_path,
          overwrite = TRUE)
name_chunks(temp_file_path)
if(interactive()){
  file.edit(temp_file_path)
}
file.remove(temp_file_path)
```

name_dir_chunks	<i>Name chunks of all Rmds in a dir</i>
-----------------	---

Description

Name unnamed chunks in a dir using the filenames with extension stripped as basis.

Usage

```
name_dir_chunks(dir)
```

Arguments

dir	Path to folder
-----	----------------

Details

When using namer, please check the edits before pushing them to your code base. Such automatic chunk labelling is best paired with version control.

Examples

```
temp_dir <- tempdir()
# just to make sure we're not overwriting
if(fs::dir_exists(file.path(temp_dir, "examples"))){
  fs::dir_delete(file.path(temp_dir, "examples"))
}
fs::dir_copy(system.file("examples", package = "namer"),
             temp_dir)
# this is an example file that'd fail
fs::file_delete(file.path(temp_dir,
                          "examples", "example4.Rmd"))
name_dir_chunks(temp_dir)
if(interactive()){
  file.edit(file.path(temp_dir,
                     "examples", "example1.Rmd"))
}
```

unname_all_chunks *Unname chunks in a single file*

Description

Unname all chunks except the setup chunk, in a single file

Usage

```
unname_all_chunks(path)
```

Arguments

path Path to file

Details

When using namer, please check the edits before pushing them to your code base. Such automatic chunk labelling is best paired with version control.

Examples

```
temp_file_path <- file.path(tempdir(), "test.Rmd")
file.copy(system.file("examples", "example4.Rmd", package = "namer"),
          temp_file_path,
          overwrite = TRUE)
unname_all_chunks(temp_file_path)
```

```
if(interactive()){  
  file.edit(temp_file_path)  
}
```

Index

`name_chunks`, [2](#)

`name_dir_chunks`, [2](#)

`unname_all_chunks`, [3](#)