

Package ‘namer’

February 29, 2024

Title Names Your 'R Markdown' Chunks

Version 0.1.8

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

License MIT + file LICENSE

URL <https://github.com/jumpingrivers/namer>,
<https://jumpingrivers.github.io/namer/>

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

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

Suggests covr, knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

RoxygenNote 7.2.3

NeedsCompilation no

Author Colin Gillespie [aut, cre],
Steph Locke [aut],
Maëlle Salmon [aut] (<<https://orcid.org/0000-0002-2815-0399>>),
Ellis Valentiner [ctb],
Charlie Hadley [ctb] (<<https://orcid.org/0000-0002-3039-6849>>),
Jumping Rivers [fnd] (<https://jumpingrivers.com>),
Han Oostdijk [ctb] (<<https://orcid.org/0000-0001-6710-4566>>),
Patrick Schratz [ctb] (<<https://orcid.org/0000-0003-0748-6624>>)

Maintainer Colin Gillespie <colin@jumpingrivers.com>

Repository CRAN

Date/Publication 2024-02-29 21:30:10 UTC

R topics documented:

name_chunks	2
name_dir_chunks	3
unname_chunks	4
unname_dir_chunks	5

Index	6
--------------	----------

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, unname = FALSE)
```

Arguments

path	Path to file
unname	Should all chunks be unnamed before naming? Default is 'FALSE'. Turning this option on will cause all existing labels to be overwritten. In contrast, with its default 'unname = FALSE' only unlabelled chunks will be named.

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.

Value

Always returns TRUE invisibly. Called for side effects.

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, unname = FALSE)
```

Arguments

dir	Path to folder
unname	Should all chunks be unnamed before naming? Default is 'FALSE'. Turning this option on will cause all existing labels to be overwritten. In contrast, with its default 'unname = FALSE' only unlabelled chunks will be named.

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.

Value

Always returns TRUE invisibly. Called for side effects.

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_chunks	<i>Unname chunks in a single file</i>
---------------	---------------------------------------

Description

Unname in a single file all chunks, or alternatively only unname the chunknames with a given prefix. In both cases, the chunk name "setup" is preserved, that chunk is never unname.

Usage

```
unname_chunks(path, chunk_name_prefix = NULL)

unname_all_chunks(path, chunk_name_prefix = NULL)
```

Arguments

path	Path to file
chunk_name_prefix	Character string with prefix of chunknames that will be removed. Default: NULL (indicating all chunknames will be removed except the one named 'setup')

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.

Value

Always returns TRUE invisibly. Called for side effects.

Examples

```
# remove all chunklabels except the one named 'setup'
temp_file_path <- file.path(tempdir(), "test1.Rmd")
file.copy(system.file("examples", "example4.Rmd", package = "namer"),
          temp_file_path,
          overwrite = TRUE)
unname_chunks(temp_file_path)
if(interactive()){
  file.edit(temp_file_path)
}
# remove all chunk labels starting with 'example4'
temp_file_path <- file.path(tempdir(), "test2.Rmd")
file.copy(system.file("examples", "example4.Rmd", package = "namer"),
          temp_file_path,
          overwrite = TRUE)
unname_chunks(temp_file_path, chunk_name_prefix='example4')
if(interactive()){
  file.edit(temp_file_path)
}
```

unname_dir_chunks	<i>Unname chunks of all Rmds in a dir</i>
-------------------	---

Description

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

Usage

```
unname_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.

Value

Always returns TRUE invisibly. Called for side effects.

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"))
}
```

Index

`name_chunks`, [2](#)

`name_dir_chunks`, [3](#)

`unname_all_chunks` (`unname_chunks`), [4](#)

`unname_chunks`, [4](#)

`unname_dir_chunks`, [5](#)