Package ‘exampletestr’

April 15, 2020

Type Package

Title Help for Writing Unit Tests Based on Function Examples

Version 1.6.1

Maintainer Rory Nolan <rorynoolan@gmail.com>

Description Take the examples written in your documentation of functions and use them to create shells (skeletons which must be manually completed by the user) of test files to be tested with the 'testthat' package. Sort of like python 'doctests' for R.

License GPL-3

URL https://rorynolan.github.io/exampletestr, https://github.com/rorynolan/exampletestr#readme

BugReports https://www.github.com/rorynolan/exampletestr/issues

Depends R (>= 3.5.0)

Imports checkmate (>= 1.9.3), filesstrings (>= 3.1.5), fs (>= 1.2.3), magrittr (>= 1.5), ore (>= 1.4.0), purrr, readr, rlang (>= 0.3.3), roxygen2, rstudioapi (>= 0.4), stringr (>= 1.3.0), styler (>= 1.2.0), usethis (>= 1.5.1), withr (>= 2.1.0)

Suggests clipr (>= 0.7.0), covr, crayon, knitr, pacman, rmarkdown, spelling, testthat (>= 2.3.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Rory Nolan [aut, cre] (<https://orcid.org/0000-0002-5239-4043>), Sergi Padilla-Parra [ths] (<https://orcid.org/0000-0002-8010-9481>), Thomas Quinn [rev] (<https://orcid.org/0000-0003-0286-6329>), Laurent Gatto [rev] (<https://orcid.org/0000-0002-1520-2268>)

Repository CRAN

Date/Publication 2020-04-15 18:10:02 UTC

R topics documented:

make-test-shells *Create test shells.*

Description

- For a given function fun() in a package, make_test_shell_fun() checks if there are examples for that function detailed in the 'man/' directory (in a '.Rd' file) and if so creates a shell (skeleton) of a testthat::test_that() test based on those examples. The created shell is then written to a corresponding file 'test-fun.R' (or if that already exists, 'test-fun-examples.R') in 'tests/testthat'.
- For a given file 'x.R' in the 'R/' directory of a package, for each function defined in that '.R' file, make_tests_shells_file() checks if there are examples for that function detailed in the 'man/' directory (in a '.Rd' file) and if so creates a shell (skeleton) of a testthat::test_that() test based on those examples via make_test_shell(). The created shells are then written to a corresponding file 'test-x.R' (or 'test-x-examples.R' if 'test-x.R' is already taken) in 'tests/testthat'.
- make_test_shells_pkg() runs make_test_shells_file() on every '.R' file in the 'R/' directory of a package.

Usage

```
make_test_shell_fun(
  fun,
  pkg_dir = ".",
  overwrite = FALSE,
  e_e = TRUE,
  roxytest = FALSE,
  open = TRUE,
  document = TRUE
)

make_tests_shells_file(
  r_file_name,
  pkg_dir = ".",
  overwrite = FALSE,
  e_e = TRUE,
  open = TRUE,
  document = TRUE
)

make_tests_shells_pkg(
  pkg_dir = ".",
  overwrite = FALSE,
  e_e = TRUE,
  open = FALSE,
  document = TRUE
)
```

Arguments

`fun` The name of the function to make a test shell for.

`pkg_dir` The directory of the R project for this package (defaults to current directory). Note that this is the parent directory of R/.

`overwrite` Overwrite if the test file you're trying to create already exists?

`e_e` Set this to `FALSE` to prevent anything from being put in the shell of a `testthat::expect_equal()` statement.

`roxytest` Copy `roxytest` package @testexamples code to clipboard instead of creating file in `tests/testthat`?

`open` Open the created test file in your editor after it is created?

`document` Run `roxygen2::roxygenize()` to update package documentation before starting?

`r_file_name` The name of the '.R' file within 'R/'. There's no need to specify the file path (as 'R/x.R', but you can do this if you want), you can just use 'x.R' for whichever file 'x' it is. You can also omit the '.R' for convenience, however using the wrong case (e.g. '.r' when the file actually has the extension '.R') will produce an error.

Value

The shell of the test file is written into tests/testthat. It has the same name as the .R file it was created from except it has "test_" tacked onto the front.

Examples

```
## Not run:
pkg_dir <- "~/mylilpkg"
usethis::create_package(pkg_dir, rstudio = FALSE, open = FALSE)
fs::file_copy(
  system.file("extdata", c("detect.R", "match.R"),
    package = "exampletestr"
  ),
  paste0(pkg_dir, "/R")
)
make_test_shell_fun("str_detect()", pkg_dir,
  document = TRUE, roxytest = TRUE
)
make_test_shell_fun("str_detect()", pkg_dir,
  document = TRUE, open = FALSE
)
make_tests_shells_file("detect", pkg_dir,
  document = FALSE, open = FALSE
)
```

```
make_tests_shells_pkg(pkg_dir,
  overwrite = TRUE, document = FALSE
)
fs::dir_delete(pkg_dir)

## End(Not run)
```

Index