Package: roxut (via r-universe)

October 19, 2026

Type Package

Title Document Unit Tests Roxygen-Style

Version 0.4.3

Date 2022-12-12

Description Much as 'roxygen2' allows one to document functions in the same file as the function itself, 'roxut' allows one to write the unit tests in the same file as the function. Once processed, the unit tests are moved to the appropriate directory. Currently supports 'testthat' and 'tinytest' frameworks. The 'roxygen2' package provides much of the infrastructure.

URL https://github.com/bryanhanson/roxut

BugReports https://github.com/bryanhanson/roxut/issues

License GPL-3

Encoding UTF-8

Depends roxygen2 (>= 7.1)

Suggests tinytest, testthat

Roxygen list(roclets = c(``namespace'', ``rd'', ``roxut::tests_roclet''))

RoxygenNote 7.2.3

Config/pak/sysreqs cmake make libuv1-dev libxml2-dev

Repository https://bryanhanson.r-universe.dev

Date/Publication 2022-12-12 23:00:04 UTC

RemoteUrl https://github.com/bryanhanson/roxut

RemoteRef HEAD

RemoteSha f9c49e0ce42f9e618315b817015337262b48bbc1

Contents

roxut-package 2
roclet_clean.roclet_tests 2
roclet_output.roclet_tests 3
roclet_process.roclet_tests 4
roxy_tag_parse.roxy_tag_tests 4
tests_roclet 5

Index **6**

`roxut-package` *roxut*

Description

Much as `roxygen2` allows one to document functions in the same file as the function itself, `roxut` allows one to write the unit tests in the same file as the function. Once processed, the unit tests are moved to the appropriate directory. Currently supports `testthat` and `tinytest` frameworks. The `roxygen2` package provides much of the infrastructure.

Author(s)

Bryan A. Hanson.

Maintainer: Bryan A. Hanson <hanson@depauw.edu>

`roclet_clean.roclet_tests`
Delete Unit Test Files Generated by @tests Tags

Description

Delete unit test files generated by @tests tags. Activated when calling `roxygenize` with `clean = TRUE`.

Usage

```
## S3 method for class 'roclet_tests'
roclet_clean(x, base_path)
```

Arguments

`x`	A `roclet` object.
`base_path`	Path to root of source package.

Value

NULL If found, files created by `roxut` will be deleted.

Author(s)

Bryan A. Hanson

`roclet_output.roclet_tests` *Write the Results of Processing @tests Tags to Unit Test File*

Description

The contents of the unit tests (lines following the `@tests` tags) are written verbatim to the appropriate directories (directory depends on whether `tinytest` or `testthat` is in use). If there is more than one function and corresponding unit tests in a file, the first unit tests will be in `filename.R`, the second in `filename-1.R` etc.

Usage

```
## S3 method for class 'roclet_tests'
roclet_output(x, results, base_path, ...)
```

Arguments

`x`	A `roclet` object.
`results`	Value returned from the `roclet_process()` method.
`base_path`	Path to root of source package.
`...`	Other arguments passed downstream. Needed for compatibility, does not appear to be used.

Value

Invisibly, a list with the text of the tests. Writes files!

Author(s)

Bryan A. Hanson

`roclet_process.roclet_tests`
Process @tests Tags

Description

Processes the blocks containing the `@tests` tags.

Usage

```
## S3 method for class 'roclet_tests'
roclet_process(x, blocks, env, base_path)
```

Arguments

`x`	A `roclet` object.
`blocks`	A list of `roxy_block` objects.
`env`	Package environment.
`base_path`	Path to root of source package.

Value

A list.

Author(s)

Bryan A. Hanson

`roxy_tag_parse.roxy_tag_tests`
Parse @tests Tags

Description

This function is called once for each `@tests` tag in a function definition file.

Usage

```
## S3 method for class 'roxy_tag_tests'
roxy_tag_parse(x)
```

Arguments

`x`	A `roclet` object.

Value

The x object is returned, appended with a list val containing two elements:

- framework containing the name of the unit test framework in use.
- tests containing the content of the unit tests.

Author(s)

Bryan A. Hanson

---

tests_roclet *tests_roclet*

Description

This roclet automates the creation of unit tests, using @tests tags written in the function definition files, and processed using roxygen2. The unit tests are written verbatim to the appropriate destination file.

Usage

```
tests_roclet()
```

Value

A complete roclet.

Author(s)

Bryan A. Hanson

Examples

```
# In actual use, the test lines would be in your function definition file.
# The results (cat'd here) would be written to the unit test file.
# If the function definition file is myFunc.R then the results
# are written to test-myFunc.R in the appropriate directory
# (the directory depends on whether your are using tinytest or testthat).

tinytest_in <- "#' @tests tinytest\n#' # Are these equal?\n#' expect_equal(5.0, 5.0)\nNULL\n"
tinytest_out <- roxygen2::roc_proc_text(tests_roclet(), tinytest_in)
cat(tinytest_out$tinytest, "\n")

testthat_in <- "#' @tests testthat\n#' # Are these equal?\n#' expect_equal(5.0, 5.0)\nNULL\n"
testthat_out <- roxygen2::roc_proc_text(tests_roclet(), testthat_in)
cat(testthat_out$testthat, "\n")
```

Index

∗ **package**
roxut-package, 2

roclet_clean.roclet_tests, 2
roclet_output.roclet_tests, 3
roclet_process.roclet_tests, 4
roxut *(*roxut-package*)*, 2
roxut-package, 2
roxy_tag_parse.roxy_tag_tests, 4

tests_roclet, 5