

Package ‘ttdo’

October 14, 2022

Title Extend 'tinytest' with 'diffobj'

Version 0.0.8

Date 2021-07-17

Author Dirk Eddelbuettel and Alton Barbehenn

Maintainer Dirk Eddelbuettel <edd@debian.org>

Description The 'tinytest' package offers a light-weight zero-dependency unit-testing framework to which this package adds support of the 'diffobj' package for 'diff'-style comparison of R objects.

License GPL (>= 2)

URL <https://github.com/eddelbuettel/ttdo/>,
<https://dirk.eddelbuettel.com/code/ttdo.html>

BugReports <https://github.com/eddelbuettel/ttdo/issues>

NeedsCompilation no

Encoding UTF-8

Imports tinytest (>= 1.3.1), diffobj

RoxygenNote 7.1.0

Repository CRAN

Date/Publication 2021-07-17 13:40:02 UTC

R topics documented:

expect_equal_with_diff	2
expect_equal_xl	3
makeDataFrame	5
ttdo_boolean_and_message_tests	6

Index	8
--------------	----------

expect_equal_with_diff

Test for equality with explicit difference

Description

Test for equality with explicit difference

Usage

```
expect_equal_with_diff(
  current,
  target,
  tol = sqrt(.Machine$double.eps),
  mode = getOption("diffobj.mode", "unified"),
  format = getOption("diffobj.format", "ansi256"),
  ...
)
```

```
expect_equivalent_with_diff(
  current,
  target,
  tol = sqrt(.Machine$double.eps),
  ...
)
```

Arguments

current	[R object or expression] Outcome or expression under scrutiny.
target	[R object or expression] Expected outcome
tol	[numeric] Test equality to machine rounding. Passed to all.equal (tolerance)
mode	[character] Comparison mode passed to <code>diffPrint</code> , defaults to using the “diffobj.mode” global option value with “unified” as fallback if no such option is set
format	[character] Comparison mode passed to <code>diffPrint</code> , defaults to using the “diffobj.format” global option value with “ansi256” as fallback if no such option is set
...	Passed to <code>all.equal</code>

Details

`expect_equivalent_with_diff` calls `expect_equal_with_diff` with the extra arguments `check.attributes=FALSE` and `use.names=FALSE`

Value

A `tinytest` object. A `tinytest` object is a logical with attributes holding information about the test that was run

Examples

```
library(tinytest)
using(ttdo)
expect_equal_with_diff(1 + 1, 2) # TRUE
expect_equal_with_diff(1 - 1, 2) # FALSE
expect_equivalent_with_diff(2, c(x=2)) # TRUE
expect_equivalent_with_diff(2, c(x=2)) # TRUE
```

 expect_equal_xl

Extensions of equality tests for tinytest

Description

Building on the `tinytest` functions for testing equality with optional enhanced object diffing and additional test feedback through additional attributes.

Usage

```
expect_equal_xl(
  current,
  target,
  useDiffObj = TRUE,
  tol = sqrt(.Machine$double.eps),
  info = NA_character_,
  mode = getOption("diffobj.mode", "unified"),
  format = getOption("diffobj.format", "ansi256"),
  ...
)

expect_identical_xl(
  current,
  target,
  useDiffObj = TRUE,
  info = NA_character_,
  mode = getOption("diffobj.mode", "unified"),
  format = getOption("diffobj.format", "ansi256"),
  ...
)

expect_equivalent_xl(
  current,
  target,
```

```

useDiffObj = TRUE,
tol = sqrt(.Machine$double.eps),
info = NA_character_,
mode = getOption("diffobj.mode", "unified"),
format = getOption("diffobj.format", "ansi256"),
...
)

```

Arguments

current	[R object or expression] Outcome or expression under scrutiny.
target	[R object or expression] Expected outcome
useDiffObj	[logical] Whether you should use diffPrint for the diff field in the resulting tinytest object
tol	[numeric] Test equality to machine rounding. Passed to <code>all.equal</code> (tolerance)
info	An additional attribute to pass around with the tinytest object
mode	[character] Comparison mode passed to diffPrint, defaults to using the “diffobj.mode” global option value with “unified” as fallback if no such option is set
format	[character] Comparison mode passed to diffPrint, defaults to using the “diffobj.format” global option value with “ansi256” as fallback if no such option is set
...	Passed to <code>all.equal</code> and returned as a test attributes

Details

While tinytest does now support the passing of additional information with the info field in its tests, they are not yet supported in the `as.data.frame.tinytests` method.

Value

A `tinytest` object. A tinytest object is a logical with attributes holding information about the test that was run

Examples

```

library(tinytest)
using(ttdo)
expect_equal_xl(1 + 1, 2, score = 3) # TRUE
expect_equal_xl(1 - 1, 2, name = "check 1-1==2", score = 1, totalpts = 2) # FALSE

```

makeDataFrame	<i>Convert tinytest results to data.frame</i>
---------------	---

Description

This method extends the `as.data.frame.tinytest` method to handle arbitrary attributes attached to each tinytest object. You can pass in the results of a single test (a tinytest object) directly or the results of one of the `run_test_*` functions (a tinytests object).

Usage

```
makeDataFrame(x)
```

Arguments

`x` a tinytest or tinytests object

Examples

```
# create a test file in tempdir
tests <- "
using(ttdo)

addOne <- function(x) x + 2

expect_true(addOne(0) > 0)
expect_equal(2, addOne(1))
"
testfile <- tempfile(pattern = "test_", fileext = ".R")
write(tests, testfile)

# extract testdir
testdir <- dirname(testfile)
# run all files starting with 'test' in testdir
library(tinytest)
out <- run_test_dir(testdir)
#
# convert results
dat <- makeDataFrame(out)
dat

dat2 <- makeDataFrame(expect_equal_xl(1-1, 2, useDiffObj = FALSE, name = 'subtr', pts = 1))
```

`ttdo_boolean_and_message_tests`*Extensions of boolean and messaging tests from tinytest*

Description

Building on the tinytest functions for testing boolean values with additional test feedback through attributes.

Usage

```
expect_true_xl(current, info = NA_character_, ...)  
expect_false_xl(current, info = NA_character_, ...)  
expect_null_xl(current, info = NA_character_, ...)  
expect_silent_xl(current, quiet = TRUE, info = NA_character_, ...)  
  
expect_error_xl(  
  current,  
  pattern = ".*",  
  class = "error",  
  info = NA_character_,  
  ...  
)  
  
expect_warning_xl(  
  current,  
  pattern = ".*",  
  class = "warning",  
  info = NA_character_,  
  ...  
)  
  
expect_message_xl(  
  current,  
  pattern = ".*",  
  class = "message",  
  info = NA_character_,  
  ...  
)
```

Arguments

`current` [R object or expression] Outcome or expression under scrutiny.

info	scalar. Optional user-defined message. Must be a single character string. Multi-line comments may be separated by "\n".
...	Passed to all .equal and returned as a test attribute
quiet	[logical] suppress output printed by the current expression (see examples)
pattern	[character] A regular expression to match the message.
class	[character] For condition signals (error, warning, message) the class from which the condition should inherit.

Details

While `tinytest` does now support the passing of additional information with the `info` field in its tests, they are not yet supported in the `as.data.frame.tinytests` method.

Value

A `tinytest` object. A `tinytest` object is a `logical` with attributes holding information about the test that was run

Examples

```
library(tinytest)
using(ttdo)
expect_true_xl(TRUE, score = 3) # TRUE
expect_true_xl(FALSE, name = "check 1-1==2", score = 1, totalpts = 2) # FALSE
```

Index

`all.equal`, [2](#), [4](#)

`expect_equal_with_diff`, [2](#)

`expect_equal_xl`, [3](#)

`expect_equivalent_with_diff`
(`expect_equal_with_diff`), [2](#)

`expect_equivalent_xl` (`expect_equal_xl`),
[3](#)

`expect_error_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_false_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_identical_xl` (`expect_equal_xl`), [3](#)

`expect_message_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_null_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_silent_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_true_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`expect_warning_xl`
(`ttdo_boolean_and_message_tests`),
[6](#)

`makeDataFrame`, [5](#)

`tinytest`, [3](#), [4](#), [7](#)

`ttdo_boolean_and_message_tests`, [6](#)