

Package: lazytest (via r-universe)

October 15, 2024

Title Runs Only Failed Tests

Version 0.0.0.9061

Description A companion package to the testthat package that saves development time by helping you rerun only the tests that have failed during the last run.

License MIT + file LICENSE

URL <https://github.com/cynkra/lazytest>, <https://lazytest.cynkra.com/>

BugReports <https://github.com/cynkra/lazytest/issues>

Imports brio, cli, fs, rex, rlang, testthat (>= 3.2.0)

Suggests callr, here, pkgload, rstudioapi, usethis, withr

Config/Needs/website cynkra/cynkratemplate

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2.9000

Repository <https://cynkra.r-universe.dev>

RemoteUrl <https://github.com/cynkra/lazytest>

RemoteRef HEAD

RemoteSha 494db762ad6a228fef66b6dcb687a86a0415a8b3

Contents

lazytest_local	2
Index	6

<code>lazytest_local</code>	<i>Test a local source package</i>
-----------------------------	------------------------------------

Description

A drop-in replacement for `testthat::test_local()` that by default reruns only the tests that failed last time.

Usage

```
lazytest_local(
  path = ".",
  reporter = NULL,
  ...,
  lazytest_reset = FALSE,
  stop_on_failure = TRUE,
  stop_on_warning = FALSE,
  filter = NULL,
  load_package = "source"
)
```

Arguments

<code>path</code>	Path to directory containing tests.
<code>reporter</code>	Reporter to use to summarise output. Can be supplied as a string (e.g. "summary") or as an R6 object (e.g. <code>SummaryReporter\$new()</code>). See Reporter for more details and a list of built-in reporters.
<code>...</code>	Additional arguments passed to <code>test_dir()</code>
<code>lazytest_reset</code>	Set to TRUE to run all tests, regardless of what the last test results were.
<code>stop_on_failure</code>	If TRUE, throw an error if any tests fail.
<code>stop_on_warning</code>	If TRUE, throw an error if any tests generate warnings.
<code>filter</code>	Must be NULL.
<code>load_package</code>	Strategy to use for load package code: <ul style="list-style-type: none"> • "none", the default, doesn't load the package. • "installed", uses <code>library()</code> to load an installed package. • "source", uses <code>pkgload::load_all()</code> to a source package. To configure the arguments passed to <code>load_all()</code>, add this field in your DESCRIPTION file: Config/testthat/load-all: <code>list(export_all = FALSE, helpers = FALSE)</code>

Value

A list (invisibly) containing data about the test results, like `testthat::test_local()`.

Example

Let's create a package with two boilerplate tests.

```
withr::local_seed(42)
withr::local_options(usethis.quiet = TRUE)

pkg_parent_dir <- withr::local_tempdir()
pkg_dir <- file.path(pkg_parent_dir, "tea")
usethis::create_package(pkg_dir)
usethis::with_project(path = pkg_dir, {
  usethis::use_testthat()
  usethis::use_test("blop")
  usethis::use_test("blip")
})
```

If we run the tests, they all pass.

```
withr::with_dir(
  pkg_dir,
  lazytest::lazytest_local()
)
#> i Testing all tests.
#> v | F W S OK | Context
#>
#> / |          0 | blip
#> v |          1 | blip
#>
#> / |          0 | blop
#> v |          1 | blop
#>
#> == Results =====
#> [ FAIL 0 | WARN 0 | SKIP 0 | PASS 2 ]
#> > Testing all tests next time.
```

Now if we replace one of the tests with a failing test,

```
brrio::write_lines(
  text = c(
    'test_that("blop() works", {',
    '  expect_equal(2 * 2, 42)',
    '})',
  ),
  path = file.path(pkg_dir, "tests", "testthat", "test-blop.R")
)
```

and then run the tests,

```

withr::with_dir(
  pkg_dir,
  lazytest::lazytest_local()
)
#> i Testing all tests.
#> v | F W S OK | Context
#>
#> / |          0 | blip
#> v |          1 | blip
#>
#> / |          0 | blop
#> - | 1          0 | blop
#> x | 1          0 | blop
#> -----
#> Failure ('test-blop.R:2:1'): blop() works
#> 2 * 2 (`actual`) not equal to 42 (`expected`).
#>
#> `actual`: 4
#> `expected`: 42
#> -----
#>
#> == Results =====
#> -- Failed tests -----
#> Failure ('test-blop.R:2:1'): blop() works
#> 2 * 2 (`actual`) not equal to 42 (`expected`).
#>
#> `actual`: 4
#> `expected`: 42
#>
#> [ FAIL 1 | WARN 0 | SKIP 0 | PASS 1 ]
#> > Testing the following tests next time:
#> * blop
#> Error: Test failures

```

a file is created with the failing test name:

```

brio::read_lines(file.path(pkg_dir, ".lazytest"))
#> [1] "blop"

```

Next time we run the tests, only this test will be run, until it is fixed at which point all tests are run again to check no failure has been introduced elsewhere.

```

withr::with_dir(
  pkg_dir,
  lazytest::lazytest_local()
)
#> i Testing only tests that failed last time:
#> * blop

```

```
#> v | F W S OK | Context
#>
#> / |          0 | blop
#> x | 1        0 | blop
#> -----
#> Failure ('test-blop.R:2:1'): blop() works
#> 2 * 2 (`actual`) not equal to 42 (`expected`).
#>
#> `actual`: 4
#> `expected`: 42
#> -----
#>
#> == Results =====
#> -- Failed tests -----
#> Failure ('test-blop.R:2:1'): blop() works
#> 2 * 2 (`actual`) not equal to 42 (`expected`).
#>
#> `actual`: 4
#> `expected`: 42
#>
#> [ FAIL 1 | WARN 0 | SKIP 0 | PASS 0 ]
#> > Repeating the same tests next time.
#> Error: Test failures
```

Index

`lazytest_local`, [2](#)

`library()`, [2](#)

`pkgload::load_all()`, [2](#)

`Reporter`, [2](#)

`test_dir()`, [2](#)

`testthat::test_local()`, [2](#)