Package: shinytesters (via r-universe)

May 24, 2026

Type Package

Title Update 'Shiny' Inputs when using testServer()

Version 0.1.0

Description Create mocked bindings to 'Shiny' update functions within test function calls to automatically update input values. The mocked bindings simulate the communication between the server and UI components of a 'Shiny' module in testServer().

URL https://ashbaldry.github.io/shinytesters/

BugReports https://github.com/ashbaldry/shinytesters/issues

License GPL-3

Encoding UTF-8

Imports cli, rlang

Suggests knitr, rmarkdown, shiny, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr

Language en-GB

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

Date/Publication 2025-08-27 08:19:55 UTC

RemoteUrl https://github.com/ashbaldry/shinytesters

RemoteRef HEAD

RemoteSha ef164d7d50372b893e08bdfb9f3b3f7447d0e77f

Contents

`create_test_update_fns` *Create Test Update Functions*

Description

Given a set of functions from an R package, create a set of mocked functions that can be used as bindings to test UI updates within `testServer`.

Usage

```
create_test_update_fns(
  fn_names,
  id_arg = "inputId",
  value_args = c("value", "selected"),
  range_value_args = c("start", "end"),
  .package = "shiny"
)
```

Arguments

`fn_names`	A character vector (string) of function names to create wrappers for
`id_arg`	A character string of the argument in ‘fn_names‘ that relates to the HTML ID argument. Default is `"inputId"`
`value_args`	A character vector of the arguments in ‘fn_names‘ that relate to the input value arguments. Defaults are ‘"value"‘ and ‘"selected‘.
`range_value_args`	A character vector of the arguments in ‘fn_names‘ that relate to the input value arguments when multiple arguments can be used to update the input. Defaults are ‘"start"‘ and ‘"end"‘.
`.package`	Character string of the package that ‘fn_names‘ exist in. Default is ‘"shiny"‘

Value

A named list of function expressions, one for each function supplied in ‘fn_names‘.

Examples

```
create_test_update_fns(
  c("updateSelectInput", "updateTextInput"),
  .package = "shiny"
)
```

use_shiny_testers *Use Shiny Testers*

Description

Enable ‘update‘ functions in the Shiny or Shiny extension package to be mocked in tests.

Usage

```
use_shiny_testers(..., .package = "shiny", .env = rlang::caller_env())

with_shiny_testers(code, ..., .package = "shiny")
```

Arguments

...	Arguments passed to create_test_update_fns
.package	Character string of the package that the update functions exist in. Default is ‘"shiny"‘
.env	Environment that defines effect scope. For expert use only.
code	Code to execute with specified bindings.

Value

Implicit return of the updated functions in the supplied package within the specified environment.

Examples

```
library(shiny)
library(testthat)

example_server_fn <- function(input, output, session) {
  observeEvent(input$trigger, {
      updateTextInput(
      inputId = "result",
      label = "New Label",
      value = NULL,
      placeholder = "New placeholder"
    )
  })
}

test_that("Check that text input gets updated", {
  use_shiny_testers()

  shiny::testServer(
    app = example_server_fn,
    expr = {
      session$setInputs(result = "Example text")
```

```
      session$setInputs(trigger = 1L)

      expect_identical(input$result, "Example text")
      expect_identical(input$result.label, "New Label")
      expect_identical(input$result.placeholder, "New placeholder")
    }
  )
})
```

Index