

Package: shinyreprex (via r-universe)

June 29, 2026

Type Package

Title Reproducible Code for 'Shiny' Objects

Version 0.2.0

Description Provides functionality to extract reactive expressions from a 'shiny' application and convert them into stand-alone R scripts. This enables users to reproduce tables and visualisations outside the interactive UI, facilitating integration into static reports or automated workflows without requiring access to the original application source code.

License MIT + file LICENSE

URL <https://github.com/AscentSoftware/shinyreprex>,
<https://ascentsoftware.github.io/shinyreprex/>

BugReports <https://github.com/AscentSoftware/shinyreprex/issues>

Depends R (>= 4.3.0)

Imports S7, constructive, purrr, rlang, styler

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

VignetteBuilder knitr

Config/Needs/website rmarkdown, ascentsoftware/acuitydown

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Collate 'Repro_S7.R' 'S7_utils.R' 'repro_call_chunk.R'
'call_chunk_function.R' 'call_chunk_generic.R'
'call_chunk_if.R' 'call_chunk_null.R' 'call_chunk_reactive.R'
'call_chunk_reactval.R' 'call_chunk_shiny.R'
'call_chunk_subset.R' 'call_chunk_switch.R'
'call_chunk_utils.R' 'repro_chunk.R' 'chunk_call.R'
'chunk_generic.R' 'chunk_reactive.R' 'package.R'
'reprex_reactive.R'

NeedsCompilation no

Author Ashley Baldry [aut, cre]

Maintainer Ashley Baldry <ashley.baldry@acuityanalytics.com>

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

Date/Publication 2026-06-29 14:31:45 UTC

RemoteUrl https://github.com/cran/shinyreprex

RemoteRef HEAD

RemoteSha 39d6f3c7453eb69e2e89a77fbd81d6df6b98728f

Contents

reprex_reactive	2
Index	4

reprex_reactive	<i>Reproduce Code</i>
-----------------	-----------------------

Description

Construct the code within a given shiny::reactive object to be able to re-create the output outside of a Shiny session.

Usage

```
reprex_reactive(x)
```

Arguments

x shiny::reactive object to make reproducible

Value

A character string, that when printed (using base::cat), displays the script that reproduces the contents of x.

Examples

```
library(shiny)

ui <- fluidPage(
  h1("Reproducible Code Example"),
  inputPanel(
    sliderInput(
      "min_width",
      "Minimum Petal Width",
```

```

      min(iris$Petal.Width),
      max(iris$Petal.Width),
      min(iris$Petal.Width),
      step = 0.1
    ),
    selectInput(
      "summary_fn",
      "Summary Function",
      c("Mean" = "mean", "Median" = "median", "SD" = "sd"),
      selected = "mean"
    )
  ),
  fluidRow(
    column(
      width = 5,
      h2("Table"),
      tableOutput("table")
    ),
    column(
      width = 7,
      h2("Code"),
      verbatimTextOutput("code")
    )
  )
)

server <- function(input, output, session) {
  iris_filt <- reactive({
    iris[with(iris, Petal.Width > input$min_width), ]
  })

  summary_tbl <- reactive({
    aggregate(
      Sepal.Width ~ Species,
      data = iris_filt(),
      FUN = get(input$summary_fn)
    )
  })

  output$table <- renderTable(summary_tbl())
  output$code <- renderText(reprex_reactive(summary_tbl))
}

if (interactive()) {
  shinyApp(ui, server)
}

```

Index

reprex_reactive, [2](#)