

# Package ‘shinyreprex’

May 9, 2026

**Type** Package

**Title** Reproducible Code for 'Shiny' Objects

**Version** 0.1.0

**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)

**Language** en-GB

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**Config/Needs/website** ascentsoftware/ascentdown

**VignetteBuilder** knitr

**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\_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** CRAN

**Date/Publication** 2026-04-02 08:00:02 UTC

## Contents

reprex_reactive . . . . .	2
<b>Index</b>	<b>4</b>

---

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](#)