

Package ‘shinyfullscreen’

July 23, 2025

Title Display 'HTML' Elements on Full Screen in 'Shiny' Apps

Version 1.1.0

Description In 'Shiny' apps, it is sometimes useful to see a plot or a table in full screen. Using 'Shinyfullscreen', you can easily designate the 'HTML' elements that can be displayed on fullscreen and use buttons to trigger the fullscreen view.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Config/testthat/edition 3

Imports shiny, jsonlite

URL <https://github.com/etiennebacher/shinyfullscreen>

BugReports <https://github.com/etiennebacher/shinyfullscreen/issues>

Suggests testthat, spelling

Language en-US

NeedsCompilation no

Author Etienne Bacher [aut, cre, cph]

Maintainer Etienne Bacher <etienne.bacher@protonmail.com>

Repository CRAN

Date/Publication 2021-01-11 20:50:03 UTC

Contents

fullscreen_all	2
fullscreen_this	3
fullscreen_those	4
Index	5

fullscreen_all	<i>Enable fullscreen for the whole page</i>
----------------	---

Description

Enable fullscreen for the whole page

Usage

```
fullscreen_all(click_id = NULL, bg_color = "#fff")
```

Arguments

click_id	Id of the item that triggers the fullscreen view. This is a mandatory argument. You can specify the id of a button for instance, so that clicking on this button triggers the fullscreen view for the whole page.
bg_color	Background color when item is displayed fullscreen. Default is white.

Value

Enables the whole page to be displayed in fullscreen mode.

Examples

```
if (interactive()) {  
  ### Only works in browser  
  
  library(shiny)  
  
  ui <- fluidPage(  
    actionButton("test", "test"),  
    plotOutput("plot"),  
    fullscreen_all(click_id = "test")  
  )  
  
  server <- function(input, output, session) {  
    output$plot <- renderPlot(plot(mtcars))  
  }  
  
  shinyApp(ui, server, options = list(launch.browser = TRUE))  
}
```

fullscreen_this	<i>Enable fullscreen for a specific item</i>
-----------------	--

Description

Enable fullscreen for a specific item

Usage

```
fullscreen_this(ui_element, click_id = NULL, bg_color = "#fff")
```

Arguments

ui_element	A UI element that should be displayed fullscreen.
click_id	Id of the item that triggers the fullscreen view. By default, it is the id of ui_element, i.e clicking on the element shows it on fullscreen. You can specify the id of a button for instance, so that clicking on this button triggers the fullscreen view of ui_element.
bg_color	Background color when item is displayed fullscreen. Default is white.

Value

Enables the selected element to be displayed in fullscreen mode.

Examples

```
if (interactive()) {  
  ### Only works in browser  
  
  library(shiny)  
  
  ui <- fluidPage(  
    actionButton("test", "test"),  
    fullscreen_this(plotOutput("plot"))  
  )  
  
  server <- function(input, output, session) {  
  
    output$plot <- renderPlot(plot(mtcars))  
  
  }  
  
  shinyApp(ui, server, options = list(launch.browser = TRUE))  
}
```

fullscreen_those	<i>Enable fullscreen for a list of items</i>
------------------	--

Description

Enable fullscreen for a list of items

Usage

```
fullscreen_those(items = list(), bg_color = "#fff")
```

Arguments

items	A list containing the ids of the items for which fullscreen is enabled.
bg_color	Background color when item is displayed full screen. Default is white.

Details

This function has to be placed AFTER the call of inputs. See Examples.

Value

Enables the selected elements to be displayed in fullscreen mode.

Examples

```
if (interactive()) {
  ### Only works in browser

  library(shiny)
  library(shinyfullscreen)

  ui <- fluidPage(
    plotOutput("plot"),
    plotOutput("plot2"),

    # Has to be placed after plot and plot2
    fullscreen_those(items = list("plot", "plot2"))
  )

  server <- function(input, output, session) {

    output$plot <- renderPlot(plot(mtcars))
    output$plot2 <- renderPlot(plot(AirPassengers))

  }

  shinyApp(ui, server, options = list(launch.browser = TRUE))
}
```

Index

fullscreen_all, [2](#)
fullscreen_this, [3](#)
fullscreen_those, [4](#)