Package ‘shinycssloaders’

October 14, 2022

Type    Package
Title   Add Loading Animations to a 'shiny' Output While It's Recalculating
Version 1.0.0

Description When a 'Shiny' output (such as a plot, table, map, etc.) is recalculating, it remains visible but gets greyed out. Using 'shinycssloaders', you can add a loading animation ("spinner") to outputs instead. By wrapping a 'Shiny' output in 'withSpinner()', a spinner will automatically appear while the output is recalculating. See the demo online at <https://daattali.com/shiny/shinycssloaders-demo/>.

License MIT + file LICENSE
URL https://github.com/daattali/shinycssloaders

BugReports https://github.com/daattali/shinycssloaders/issues

Depends R (>= 3.1)
Imports digest, glue, grDevices, shiny
Suggests shinydisconnect, shinyjs

RoxygenNote 7.1.0
Encoding UTF-8

NeedsCompilation no

Author Andras Sali [aut] (Original creator of shinycssloaders package), Luke Hass [ctb, cph] (Author of included CSS loader code), Dean Attali [aut, cre] (Maintainer of shinycssloaders since 2019)

Maintainer Dean Attali <daattali@gmail.com>
Repository CRAN

Date/Publication 2020-07-28 16:20:03 UTC

R topics documented:

  shinycssloaders .......................................................... 2
  withSpinner .............................................................. 2

Index 4
Description

Add automatic loading animations to your Shiny outputs (e.g. plots, tables) by adding a single function call around the Shiny output function. Animations automatically show/hide when the output is recalculating/recalculated. You can use your own custom image, or choose from 8 different types of built-in animations, as well customize the color and size of the built-in animations.

Details

For further reference on how to use the package, please refer to https://github.com/daattali/shinycssloaders

Usage

```
withSpinner(
  ui_element,
  type = getOption("spinner.type", default = 1),
  color = getOption("spinner.color", default = "+0275D8"),
  size = getOption("spinner.size", default = 1),
  color.background = getOption("spinner.color.background"),
  custom.css = FALSE,
  proxy.height = NULL,
  id = NULL,
  image = NULL,
  image.width = NULL,
  image.height = NULL,
  hide.ui = TRUE
)
```

Arguments

- `ui_element` A UI element that should be wrapped with a spinner when the corresponding output is being calculated.
withSpinner

- **type**
  The type of spinner to use, valid values are integers between 0-8 (0 means no spinner). Check out [https://daattali.com/shiny/shinycssloaders-demo](https://daattali.com/shiny/shinycssloaders-demo) to see the different types of spinners. You can also use your own custom image using the `image` parameter.

- **color**
  The color of the spinner in hex format. Ignored if `image` is used.

- **size**
  The size of the spinner, relative to its default size (default is 1, a size of 2 means twice as large). Ignored if `image` is used.

- **color.background**
  For certain spinners (type 2-3), you will need to specify the background color of the spinner. Ignored if `image` is used.

- **custom.css**
  Set to TRUE if you have your own custom CSS that you defined and you don’t want the automatic CSS applied to the spinner. Ignored if `image` is used.

- **proxy.height**
  If the output UI doesn’t specify the output height, you can set a proxy height. It defaults to "400px" for outputs with undefined height. Ignored if `hide.ui` is set to FALSE.

- **id**
  The HTML ID to use for the spinner. If you don’t provide one, it will be generated automatically.

- **image**
  The path or URL of the image to use if you want to use a custom image instead of a built-in spinner.

- **image.width**
  The width for the custom image spinner, in pixels. If not provided, then the original size of the image is used. Ignored if not using `image`.

- **image.height**
  The height for the custom image spinner, in pixels. If not provided, then the original size of the image is used. Ignored if not using `image`.

- **hide.ui**
  By default, while an output is recalculating, the output UI is hidden and the spinner is visible instead. Setting `hide.ui = FALSE` will result in the spinner showing up on top of the previous output UI.

**Examples**

```r
if (interactive()) {
  library(shiny)

  shinyApp(
    ui = fluidPage(
      actionButton("go", "Go"),
      withSpinner(plotOutput("plot"))
    ),
    server = function(input, output) {
      output$plot <- renderPlot(
        input$go,
        Sys.sleep(1.5),
        plot(runif(10))
      )
    }
  )
}
```
Index

shinycssloaders, 2
withSpinner, 2