

# Package ‘shinycssloaders’

July 31, 2024

**Title** Add Loading Animations to a 'shiny' Output While It's Recalculating

**Version** 1.1.0

**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. You can also manually show and hide the spinner, or add a full-page spinner to cover the entire page.  
See the demo online at <<https://daattali.com/shiny/shinycssloaders-demo/>>.

**License** MIT + file LICENSE

**URL** <https://github.com/daattali/shinycssloaders>,  
<https://daattali.com/shiny/shinycssloaders-demo/>

**BugReports** <https://github.com/daattali/shinycssloaders/issues>

**Depends** R (>= 3.1)

**Imports** digest, glue, grDevices, htmltools (>= 0.3.5), shiny

**Suggests** knitr, shinydisconnect, shinyjs

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Dean Attali [aut, cre] (Maintainer/developer of shinycssloaders since 2019, <<https://orcid.org/0000-0002-5645-3493>>),  
Andras Sali [aut] (Original creator of shinycssloaders package),  
Luke Hass [ctb, cph] (Author of included CSS loader code)

**Maintainer** Dean Attali <daattali@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-07-30 22:30:02 UTC

## Contents

|                        |   |
|------------------------|---|
| showHide . . . . .     | 2 |
| showHidePage . . . . . | 3 |
| withSpinner . . . . .  | 5 |

---

|          |                                     |
|----------|-------------------------------------|
| showHide | <i>Manually show/hide a spinner</i> |
|----------|-------------------------------------|

---

### Description

Any Shiny output that uses `withSpinner()` will automatically show a spinner while it's recalculating. Use `showSpinner()` and `hideSpinner()` to manually trigger the spinner on-demand.

### Usage

```
showSpinner(id, expr)
```

```
hideSpinner(id)
```

### Arguments

|                   |  |
|-------------------|--|
| <code>id</code>   | The ID of the Shiny output. The corresponding output must have been wrapped in <code>withSpinner()</code> in the UI.                   |
| <code>expr</code> | (optional) An R expression to run while showing the spinner. The spinner will automatically get hidden when this expression completes. |

### Value

If `expr` is provided, the result of `expr` is returned. Otherwise, `NULL`.

### See Also

[withSpinner\(\)](#)

### Examples

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

  #--- Example 1: Using showSpinner/hideSpinner ---

  shinyApp(
    ui = fluidPage(
      actionButton("show", "Show"),
      actionButton("hide", "Hide"),
      withSpinner(plotOutput("plot"))
    ),
    server = function(input, output) {
      output$plot <- renderPlot({
        plot(runif(10))
      })
      observeEvent(input$show, {
        showSpinner("plot")
      })
    }
  )
}
```

```

    })
    observeEvent(input$hide, {
      hideSpinner("plot")
    })
  }
)

#--- Example 2: Using showSpinner with expr ---

some_slow_function <- function() {
  Sys.sleep(2)
}

shinyApp(
  ui = fluidPage(
    actionButton("show", "Show"),
    withSpinner(plotOutput("plot"))
  ),
  server = function(input, output) {
    output$plot <- renderPlot({
      plot(runif(10))
    })
    observeEvent(input$show, {
      showSpinner("plot", { some_slow_function() })
    })
  }
)
}

```

---

showHidePage

*Show (and hide) a full-page spinner that covers the entire page*


---

## Description

Use these functions to show and hide a full-page spinner.

All parameters (except `expr`) can be set globally in order to use a default setting for all full-page spinner in your Shiny app. This can be done by setting an R option with the parameter's name prepended by `"page.spinner."`. For example, to set all page spinners to `type=5` and `color=#0dc5c1` by default, use `options(page.spinner.type = 5, page.spinner.color = "#0dc5c1")`.

## Usage

```

showPageSpinner(
  expr,
  background = getOption("page.spinner.background", default = "#FFFFFFCC"),
  type = getOption("page.spinner.type", default = 8),
  color = getOption("page.spinner.color", default = "#0275D8"),

```

```

size = getOption("page.spinner.size", default = 1),
color.background = getOption("page.spinner.color.background"),
custom.css = getOption("page.spinner.custom.css", default = FALSE),
id = getOption("page.spinner.id"),
image = getOption("page.spinner.image"),
image.width = getOption("page.spinner.image.width"),
image.height = getOption("page.spinner.image.height"),
caption = getOption("page.spinner.caption")
)

hidePageSpinner()

```

### Arguments

|                               |  |
|-------------------------------|--|
| <code>expr</code>             | (optional) An R expression to run while showing the spinner. The spinner will automatically get hidden when this expression completes. If not provided, you must explicitly end the spinner with a call to <code>hidePageSpinner()</code> .  |
| <code>background</code>       | Background color for the spinner. You can use semi-transparent colours in order to have the Shiny app visible in the background, eg. <code>"#FFFFFFD0"</code> or <code>"rgba(0, 0, 0, 0.7)"</code> .   |
| <code>type</code>             | The type of spinner to use. Valid values are integers between 0-8 (0 means no spinner). Check out <a href="https://daattali.com/shiny/shinycssloaders-demo/">https://daattali.com/shiny/shinycssloaders-demo/</a> to see the different types of spinners. You can also use your own custom image using the <code>image</code> parameter. |
| <code>color</code>            | The color of the spinner in hex format. Ignored if <code>image</code> is used.   |
| <code>size</code>             | The size of the spinner, relative to its default size (default is 1, a size of 2 means twice as large). Ignored if <code>image</code> is used.   |
| <code>color.background</code> | For certain spinners (type 2-3), you will need to specify the background color of the spinner. Ignored if <code>image</code> is used.  |
| <code>custom.css</code>       | 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 <code>image</code> is used.   |
| <code>id</code>               | The HTML ID to use for the spinner. If you don't provide one, it will be generated automatically.  |
| <code>image</code>            | The path or URL of the image to use if you want to use a custom image instead of a built-in spinner. If <code>image</code> is provided, then <code>type</code> is ignored.   |
| <code>image.width</code>      | 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 <code>image</code> .   |
| <code>image.height</code>     | 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 <code>image</code> .  |
| <code>caption</code>          | Caption to display below the spinner or image (text or HTML). The caption's font color is determined by the <code>color</code> parameter. Ignored if <code>type</code> is 1.   |

### Value

If `expr` is provided, the result of `expr` is returned. Otherwise, NULL.

**See Also**

[withSpinner\(\)](#), [showSpinner\(\)](#), [hideSpinner\(\)](#)

**Examples**

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

  #--- Example 1: Using showPageSpinner/hidePageSpinner ---

  ui <- fluidPage(
    actionButton("go", "Go"),
    plotOutput("plot")
  )
  server <- function(input, output) {
    observeEvent(input$go, {
      showPageSpinner()
      Sys.sleep(1)
      hidePageSpinner()
    })
    output$plot <- renderPlot({
      plot(runif(10))
    })
  }
  shinyApp(ui, server)

  #--- Example 2: Using showPageSpinner with expr ---

  some_slow_function <- function() {
    Sys.sleep(1)
  }

  ui <- fluidPage(
    actionButton("go", "Go"),
    plotOutput("plot")
  )
  server <- function(input, output) {
    observeEvent(input$go, {
      showPageSpinner({ some_slow_function() })
    })
    output$plot <- renderPlot({
      plot(runif(10))
    })
  }
  shinyApp(ui, server)
}
```

## Description

Add a spinner that automatically shows while an output is recalculating. You can also manually trigger the spinner using `showSpinner()` and `hideSpinner()`.

Most parameters can be set globally in order to use a default setting for all spinners in your Shiny app. This can be done by setting an R option with the parameter's name prepended by "spinner.". For example, to set all spinners to type=5 and color=#0dc5c1 by default, use `options(spinner.type = 5, spinner.color = "#0dc5c1")`. The following parameters cannot be set globally: `ui_element`, `id`.

Use `showPageSpinner()` to show a spinner on the entire page instead of individual outputs.

## 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 = getOption("spinner.custom.css", default = FALSE),
  proxy.height = getOption("spinner.proxy.height"),
  id = NULL,
  image = getOption("spinner.image"),
  image.width = getOption("spinner.image.width"),
  image.height = getOption("spinner.image.height"),
  hide.ui = getOption("spinner.hide.ui", default = TRUE),
  caption = getOption("spinner.caption")
)
```

## Arguments

|                               |  |
|-------------------------------|--|
| <code>ui_element</code>       | A UI element that should be wrapped with a spinner when the corresponding output is being calculated.  |
| <code>type</code>             | The type of spinner to use. Valid values are integers between 0-8 (0 means no spinner). Check out <a href="https://daattali.com/shiny/shinycssloaders-demo/">https://daattali.com/shiny/shinycssloaders-demo/</a> to see the different types of spinners. You can also use your own custom image using the <code>image</code> parameter. |
| <code>color</code>            | The color of the spinner in hex format. Ignored if <code>image</code> is used.   |
| <code>size</code>             | The size of the spinner, relative to its default size (default is 1, a size of 2 means twice as large). Ignored if <code>image</code> is used.   |
| <code>color.background</code> | For certain spinners (type 2-3), you will need to specify the background color of the spinner. Ignored if <code>image</code> is used.  |
| <code>custom.css</code>       | 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 <code>image</code> 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. If image is provided, then type is ignored.  |
| 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. |
| caption      | Caption to display below the spinner or image (text or HTML). The caption's font color is determined by the color parameter. Ignored if type is 1.  |

### See Also

[showSpinner\(\)](#), [hideSpinner\(\)](#), [showPageSpinner\(\)](#)

### Examples

```
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

hidePageSpinner (showHidePage), 3  
hideSpinner (showHide), 2  
hideSpinner(), 2, 5–7

showHide, 2  
showHidePage, 3  
showPageSpinner (showHidePage), 3  
showPageSpinner(), 6, 7  
showSpinner (showHide), 2  
showSpinner(), 2, 5–7

withSpinner, 5  
withSpinner(), 2, 5