

Package: shinycustomloader (via r-universe)

July 5, 2026

Type Package

Title Custom Loader for Shiny Outputs

Version 0.9.0

Maintainer Emi Tanaka <emi.tanaka@sydney.edu.au>

Description A custom css/html or gif/image file for the loading screen in R 'shiny'. It also can use the marquee to have custom text loading screen.

License GPL-3

Encoding UTF-8

LazyData true

Imports glue, shiny

Suggests shinycssloaders

RoxygenNote 6.0.1

BugReports <https://github.com/emitanaka/shinycustomloader/issues>

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository <https://emitanaka.r-universe.dev>

Date/Publication 2018-07-17 18:50:06 UTC

RemoteUrl <https://github.com/emitanaka/shinycustomloader>

RemoteRef HEAD

RemoteSha 3131993da88e999fdb9968d686e422a1892964b8

Contents

marquee	2
renderCSS	2
shinycustomloader	3
shinyExample	3
withLoader	4

Index	5
--------------	----------

marquee	<i>Create a marquee list.</i>
---------	-------------------------------

Description

This creates the necessary elements for marquee to make custom text loading screen. This function is to be used as input list in withLoader or renderCSS functions.

Usage

```
marquee(content, behavior = "scroll", direction = "left",
        scrollamount = 6, width = "100%", ...)
```

Arguments

content	The text content of the marquee.
behavior	Sets how the text is scrolled within the marquee. Possible values are scroll, slide and alternate. If no value is specified, the default value is scroll.
direction	Sets the direction of the scrolling within the marquee. Possible values are left, right, up and down. If no value is specified, the default value is left.
scrollamount	Sets the amount of scrolling at each interval in pixels. The default value is 6.
width	Sets the width in pixels or percentage value.
...	Other parameters passed to the marquee such as height, bgcolor.

Examples

```
marquee("Custom loading text here...", height=60, width=100)
```

renderCSS	<i>Render a permanent loading screen.</i>
-----------	---

Description

This function follows similar to withLoader but needs no UI element. This allows for a permanent loading screen.

Usage

```
renderCSS(type, loader)
```

Arguments

type	The type of loader to use. Possible values are image, html or text.
loader	The name of the loader. The built in options are dnaspin, dancingtree, pacman and walkingcow. For custom html option, the name of the css and html file have to be the same and input must be without the extension. For custom text option, loader must be a list of marquee objects.

Examples

```
## Not run: marquee_list <- list(marquee("Your custom text here"))  
## Not run: renderCSS(type="text", loader=marquee_list)
```

shinycustomloader	<i>shinycustomloader</i>
-------------------	--------------------------

Description

This package extends the shinycssloaders R-package to allow for custom css animations or external gifs for loading animations by wrapping a function around the shiny output. The package contains built-in css animations such as a spinning DNA animation, walking cow animation and a pacman animation.

Details

For further reference see <https://github.com/emitanaka/shinycustomloader>.

shinyExample	<i>Example Shiny App</i>
--------------	--------------------------

Description

Shows an example shiny app with the built-in load screens.

Usage

```
shinyExample()
```

Examples

```
## Not run: shinyExample()
```

withLoader *Add a loader that shows when an output is recalculating*

Description

Add a loader that shows when an output is recalculating

Usage

```
withLoader(ui_element, type = "html", loader = "dnaspin",
  proxy.height = if (grepl("height:\\s*\\d", ui_element)) NULL else
  "400px")
```

Arguments

ui_element	A UI element that should be wrapped with a loader when the corresponding output is being calculated.
type	The type of loader to use. Possible values are image, html or text.
loader	The name of the loader. The built in options are dnaspin, dancingtree, pacman and walkingcow. For custom html option, the name of the css and html file have to be the same and input must be without the extension. For custom image option, the name must include the extension of the file. For custom text option, loader must be a list of marquee objects.
proxy.height	If the output doesn't specify the output height, you can set a proxy height. It defaults to 400px for outputs with undefined height.

References

‘shinycssloaders’ <https://github.com/andrewsali/shinycssloaders>
‘dnaspin’ <https://codepen.io/jsnainigans/pen/ORNJNp>
‘pacman’ <https://codepen.io/wifi/pen/oIKxE>
‘loaders’ <https://codepen.io/Manoz/pen/pydxK>

Examples

```
## Not run: withLoader(plotOutput("my_plot"))
## Not run: marquee_list <- list(marquee("Your custom text here"))
## Not run: withLoader(plotOutput("distPlot"), type="text", loader=marquee_list)
```

Index

marquee, [2](#)

renderCSS, [2](#)

shinycustomloader, [3](#)

shinycustomloader-package
 (shinycustomloader), [3](#)

shinyExample, [3](#)

withLoader, [4](#)