

Package ‘shinyFeedback’

August 20, 2018

Type Package

Title Displays User Feedback Next to Shiny Inputs

Version 0.1.0

Date 2018-08-19

Description Easily display user feedback next to Shiny inputs. The feedback message is displayed when the feedback condition evaluates to TRUE.

License GPL-3

LazyData true

Depends R (>= 3.1.2)

RoxygenNote 6.1.0

Suggests knitr, rmarkdown

VignetteBuilder knitr

Imports shiny, digest

NeedsCompilation no

Author Andy Merlino [aut, cre]

Maintainer Andy Merlino <merlinoa88@gmail.com>

Repository CRAN

Date/Publication 2018-08-20 04:40:02 UTC

R topics documented:

feedback	2
feedbackDanger	3
feedbackSuccess	4
feedbackWarning	5
showSnackBar	6
snackbar	7
snackbarDanger	8
snackbarSuccess	8
snackbarWarning	9
useShinyFeedback	9

feedback	<i>feedback</i>
----------	-----------------

Description

displays feedback next to Shiny input

Usage

```
feedback(inputId, condition, text = NULL, color = NULL, icon = NULL)
```

Arguments

inputId	the Shiny input's inputId argument
condition	condition under which feedback is displayed
text	text string to display below input
color	the color of the feedback
icon	a shiny::icon object

See Also

[feedbackWarning](#), [feedbackDanger](#), [feedbackSuccess](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
    useShinyFeedback(),

    numericInput(
      "exampleInput",
      "Show Feedback When < 0",
      value = -5
    )
  )

  server <- function(input, output) {
    observeEvent(input$exampleInput, {
      feedback(
        "exampleInput",
        condition = input$exampleInput < 0,
        text = "I am negative",
        color = "#d9534f",
        icon = shiny::icon("exclamation-sign", lib="glyphicon")
      )
    })
  }
}
```

```
    })
  }

  shinyApp(ui, server)
}
```

feedbackDanger	<i>feedbackDanger</i>
----------------	-----------------------

Description

wrapper for `feedback()` function that displays a danger message by default

Usage

```
feedbackDanger(inputId, condition, text = "Danger, turn back!",
  color = "#d9534f", icon = shiny::icon("exclamation-sign", lib =
  "glyphicon"))
```

Arguments

<code>inputId</code>	the Shiny input's <code>inputId</code> argument
<code>condition</code>	condition under which feedback is displayed
<code>text</code>	default "Danger, turn back" text string to display below input
<code>color</code>	default "#d9534f" the color of the feedback
<code>icon</code>	default <code>icon("exclamation-sign", lib="glyphicon")</code> a <code>shiny::icon</code> object

See Also

[feedback](#), [feedbackWarning](#), [feedbackSuccess](#)

Other feedback wrappers: [feedbackSuccess](#), [feedbackWarning](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
    useShinyFeedback(),

    numericInput(
      "exampleInput",
      "Show Feedback When < 0",
      value = -5
    )
  )
}
```

```

)

server <- function(input, output) {
  observeEvent(input$exampleInput, {
    feedbackDanger(
      "exampleInput",
      condition = input$exampleInput < 0
    )
  })
}

shinyApp(ui, server)
}

```

feedbackSuccess

feedbackSuccess

Description

wrapper for `feedback()` function that displays a success message by default

Usage

```
feedbackSuccess(inputId, condition, text = NULL, color = "#5cb85c",
  icon = shiny::icon("ok", lib = "glyphicon"))
```

Arguments

<code>inputId</code>	the Shiny input's <code>inputId</code> argument
<code>condition</code>	condition under which feedback is displayed
<code>text</code>	default NULL text string to display below input
<code>color</code>	default "#5cb85c" the color of the feedback
<code>icon</code>	default <code>icon("ok", lib="glyphicon")</code> a <code>shiny::icon</code> object

See Also

[feedback](#), [feedbackDanger](#), [feedbackSuccess](#)

Other feedback wrappers: [feedbackDanger](#), [feedbackWarning](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
    useShinyFeedback(),

```

```
    numericInput(  
      "exampleInput",  
      "Show Feedback When < 0",  
      value = -5  
    )  
  )  
  
  server <- function(input, output) {  
    observeEvent(input$exampleInput, {  
      feedbackSuccess(  
        "exampleInput",  
        condition = input$exampleInput < 0  
      )  
    })  
  }  
  
  shinyApp(ui, server)  
}
```

feedbackWarning

feedbackWarning

Description

wrapper for `feedback()` function that displays a warning message by default

Usage

```
feedbackWarning(inputId, condition, text = "Ye be warned",  
  color = "#F89406", icon = shiny::icon("warning-sign", lib =  
  "glyphicon"))
```

Arguments

<code>inputId</code>	the Shiny input's <code>inputId</code> argument
<code>condition</code>	condition under which feedback is displayed
<code>text</code>	default "Ye be warned" text string to display below input
<code>color</code>	default "#F89406" the color of the feedback
<code>icon</code>	default <code>icon("warning-sign", lib="glyphicon")</code> a <code>shiny::icon</code> object

See Also

[feedback](#), [feedbackDanger](#), [feedbackSuccess](#)

Other feedback wrappers: [feedbackDanger](#), [feedbackSuccess](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
    useShinyFeedback(),

    numericInput(
      "exampleInput",
      "Show Feedback When < 0",
      value = -5
    )
  )

  server <- function(input, output) {
    observeEvent(input$exampleInput, {
      feedbackWarning(
        "exampleInput",
        condition = input$exampleInput < 0
      )
    })
  }

  shinyApp(ui, server)
}
```

showSnackBar

showSnackBar

Description

Server side function to show a snackbar. This function should be called in the expression passed to `shiny::observe()` or `shiny::observeEvent()`.

Usage

```
showSnackBar(id, autoHideDuration = 3000)
```

Arguments

`id` A length 1 character vector. A unique id for the snackbar.

`autoHideDuration` A length 1 numeric vector. The amount of time in milliseconds to show the snackbar (e.g. 3000 is 3 seconds). Set to NULL to keep snackbar open indefinitely.

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- fluidPage(
    useShinyFeedback(),

    actionButton(
      "showSnackbarBtn",
      "Show Snackbar"
    ),

    snackbar(
      id = "mySnackbar",
      message = "Have a snack!"
    )
  )

  server <- function(input, output) {
    observeEvent(input$showSnackbarBtn, {
      showSnackbar("mySnackbar")
    })
  }

  shinyApp(ui, server)
}
```

snackbar

snackbar

Description

UI side function to generate the snackbar html.

Usage

```
snackbar(id, message, includeRemoveButton = TRUE, class = "",
  style = "")
```

Arguments

<code>id</code>	A length 1 character vector. The unique id of the snackbar. This needs to match up with the <code>id</code> argument specified in the server side call to <code>showSnackbar()</code> .
<code>message</code>	the text message to display in the snackbar
<code>includeRemoveButton</code>	whether or not to show the remove button on the right side of the snackbar
<code>class</code>	A length 1 character vector of CSS classes to add to the snackbar.
<code>style</code>	A length 1 character vector of CSS styles to add to the snackbar.

See Also[showSnackbar](#)

snackbarDanger	<i>snackbarDanger</i>
----------------	-----------------------

Description

This is a wrapper for [snackbar](#). It provides red danger/error styling for the snackbar.

Usage

```
snackbarDanger(id, message, includeRemoveButton = TRUE)
```

Arguments

id	A length 1 character vector. The unique id of the snackbar. This needs to match up with the id argument specified in the serve side call to <code>showSnackbar()</code> .
message	the text message to display in the snackbar
includeRemoveButton	whether or not to show the remove button on the right side of the snackbar

See Also[showSnackbar](#)Other snackbar wrappers: [snackbarSuccess](#), [snackbarWarning](#)

snackbarSuccess	<i>snackbarSuccess</i>
-----------------	------------------------

Description

This is a wrapper for [snackbar](#). It provides green success styling for the snackbar.

Usage

```
snackbarSuccess(id, message, includeRemoveButton = TRUE)
```

Arguments

id	A length 1 character vector. The unique id of the snackbar. This needs to match up with the id argument specified in the serve side call to <code>showSnackbar()</code> .
message	the text message to display in the snackbar
includeRemoveButton	whether or not to show the remove button on the right side of the snackbar

See Also[showSnackbar](#)Other snackbar wrappers: [snackbarDanger](#), [snackbarWarning](#)

snackbarWarning	<i>snackbarWarning</i>
-----------------	------------------------

Description

This is a wrapper for [snackbar](#). It provides yellow warning styling for the snackbar.

Usage

```
snackbarWarning(id, message, includeRemoveButton = TRUE)
```

Arguments

id	A length 1 character vector. The unique id of the snackbar. This needs to match up with the id argument specified in the server side call to <code>showSnackbar()</code> .
message	the text message to display in the snackbar
includeRemoveButton	whether or not to show the remove button on the right side of the snackbar

See Also[showSnackbar](#)Other snackbar wrappers: [snackbarDanger](#), [snackbarSuccess](#)

useShinyFeedback	<i>useShinyFeedback</i>
------------------	-------------------------

Description

function to load js for using shinyFeedback

Usage

```
useShinyFeedback()
```

Index

feedback, [2](#), [3–5](#)
feedbackDanger, [2](#), [3](#), [4](#), [5](#)
feedbackSuccess, [2–4](#), [4](#), [5](#)
feedbackWarning, [2–4](#), [5](#)

showSnackbar, [6](#), [8](#), [9](#)
snackbar, [7](#), [8](#), [9](#)
snackbarDanger, [8](#), [9](#)
snackbarSuccess, [8](#), [8](#), [9](#)
snackbarWarning, [8](#), [9](#), [9](#)

useShinyFeedback, [9](#)