Package 'shinyBS'

July 23, 2025

Type Package

Title Twitter Bootstrap Components for Shiny

Version 0.61.1

Date 2015-03-30

Author Eric Bailey

Maintainer Eric Bailey <ebailey@idem.in.gov>

Description Adds additional Twitter Bootstrap components to Shiny.

Imports shiny (>= 0.11), htmltools

URL https://ebailey78.github.io/shinyBS

BugReports https://github.com/ebailey78/shinyBS/issues

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-04-17 17:43:50 UTC

Contents

`addPopover` *addPopover*

Description

`addPopover` is used within the Server logic of an app to add a popover to a Shiny input or output.

Usage

```
addPopover(session, id, title, content, placement = "bottom",
  trigger = "hover", options = NULL)
```

Arguments

`session`	The session object passed to function given to shinyServer.
`id`	The id of the element to attach the popover to.
`title`	The title of the popover.
`content`	The main content of the popover.
`placement`	Where the popover should appear relative to its target (`top`, `bottom`, `left`, or `right`). Defaults to `bottom`.
`trigger`	What action should cause the popover to appear? (`hover`, `focus`, `click`, or `manual`). Defaults to `hover`.
`options`	A named list of additional options to be set on the popover.

Details

See Tooltips_and_Popovers for more information about how to use `addPopover` with the rest of the Tooltips_and_Popovers family.

Note

Run `bsExample("Tooltips_and_Popovers")` for an example of `addPopover` functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: Tooltips_and_Popovers; addTooltip; bsPopover; bsTooltip; popify; removePopover; removeTooltip; tipify

---

addTooltip *addTooltip*

Description

addTooltip is used within the Server logic of an app to add a tooltip to a Shiny input or output.

Usage

```
addTooltip(session, id, title, placement = "bottom", trigger = "hover",
  options = NULL)
```

Arguments

session	The session object passed to function given to shinyServer.
id	The id of the element to attach the tooltip to.
title	The content of the tooltip.
placement	Where the tooltip should appear relative to its target (top, bottom, left, or right). Defaults to "bottom".
trigger	What action should cause the tooltip to appear? (hover, focus, click, or manual). Defaults to "hover".
options	A named list of additional options to be set on the tooltip.

Details

See Tooltips_and_Popovers for more information about how to use addTooltip with the rest of the Tooltips_and_Popovers family.

Note

Run bsExample("Tooltips_and_Popovers") for an example of addTooltip functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: Tooltips_and_Popovers; addPopover; bsPopover; bsTooltip; popify; removePopover; removeTooltip; tipify

Alerts *Alerts*

Description

Alerts allow you to communicate information to the user on the fly. Standard Bootstrap styling options give the user a hint at the type of information contained in the Alert.

Details

To create alerts in your Shiny app you must place `bsAlert` in your ui. This serves as an anchor that tells shinyBS where to place the alerts created with `createAlert`.

Use `createAlert` in your server script to add alerts to the anchor you created with `bsAlert` in your ui. You can place `createAlert` in observers, reactives, or outputs. A common usage may be to have logic that validates a user's inputs. If they are valid produce the requested output, if not use `createAlert` to give the user info about what they need to change.

Components

There are three functions in the Alerts family:

`bsAlert` Used in the UI to create an anchor where your Alerts will be displayed.

`createAlert` Used in the Server logic to create alerts. This would be used within a reactive context to display error or success messages to the user based on the status of that context.

`closeAlert` Used in the Server logic to close an alert that is already open. By default, Alerts are dismissable by the user, but this offers you a way to close them programmatically.

Changes

`style` was called `type` in previous versions of shinyBS.

`anchorId` was called `inputId` in previous versions of shinyBS.

`content` was called `message` in previous versions of shinyBS.

Note

Run `bsExample("Alerts")` for an example of `Alerts` functionality.

See Also

Twitter Bootstrap 3

Other Alerts: `bsAlert`; `closeAlert`; `createAlert`

Examples

```
library(shiny)
library(shinyBS)
app = shinyApp(
 ui =
   fluidPage(
     sidebarLayout(
       sidebarPanel(textInput("num1", NULL, value = 100),
         "divided by", textInput("num2", NULL, value = 20),
         "equals", textOutput("exampleOutput")),
       mainPanel(
         bsAlert("alert")
       )
     )
 ),
 server =
   function(input, output, session) {
     output$exampleOutput <- renderText({
       num1 <- as.numeric(input$num1)
       num2 <- as.numeric(input$num2)

       if(is.na(num1) | is.na(num2)) {
         createAlert(session, "alert", "exampleAlert", title = "Oops",
           content = "Both inputs should be numeric.", append = FALSE)
       } else if(num2 == 0) {
         createAlert(session, "alert", "exampleAlert", title = "Oops",
           content = "You cannot divide by 0.", append = FALSE)
       } else {
         closeAlert(session, "exampleAlert")
         return(num1/num2)
       }

     })
   }
)

## Not run:
 runApp(app)

## End(Not run)
```

---

bsAlert *bsAlert*

Description

bsAlert creates an anchor point in your UI definition. This anchor point is where alerts created in your Server logic will be displayed.

Usage

```
bsAlert(anchorId)
```

Arguments

anchorId	A unique id the identifies the anchor.

Details

See Alerts for more information about how to use bsAlert with the rest of the Alerts family.

Note

Run bsExample("Alerts") for an example of bsAlert functionality.

See Also

Twitter Bootstrap 3

Other Alerts: Alerts; closeAlert; createAlert

---

bsButton *bsButton*

Description

bsButton is used in your UI script to create customizable action and toggle buttons.

Usage

```
bsButton(inputId, label, icon = NULL, ..., style = "default",
  size = "default", type = "action", block = FALSE, disabled = FALSE,
  value = FALSE)
```

Arguments

inputId	Specifies the input slot that will be used to access the value.
label	The contents of the button or link–usually a text label, but you could also use any other HTML, like an image.
icon	An optional icon to appear on the button.
...	Named attributes to be applied to the button or link.
style	A Bootstrap style to apply to the button. (default, primary, success, info, warning, or danger)
size	The size of the button (extra-small, small, default, or large)
type	The type of button to create. (action or toggle)
block	**logical** Should the button take the full width of the parent element?
disabled	**logical** Should the button be disabled (un-clickable)?
value	**logical** If type = "toggle", the initial value of the button.

Details

See Buttons for more information about how to use bsButton with the rest of the Buttons family.

Note

Run bsExample("Buttons") for an example of bsButton functionality.

See Also

Twitter Bootstrap 3

Other Buttons: Buttons; updateButton

---

bsCollapse *bsCollapse*

Description

bsCollapse is used in your UI to create a collapse panel group. Use bsCollapsePanel to populate this object with panels.

Usage

```
bsCollapse(..., id = NULL, multiple = FALSE, open = NULL)
```

Arguments

id	**Optional** You can use input$id in your Server logic to determine which panels are open, and updateCollapse to open/close panels.
multiple	Can more than one panel be open at a time? Defaults to FALSE.
open	The value, (or if none was supplied, the title) of the panel(s) you want open on load.
...	bsCollapsePanel elements to include in the Collapse.

Details

See Collapses for more information about how to use bsCollapse with the rest of the Collapses family.

Note

Run bsExample("Collapses") for an example of bsCollapse functionality.

See Also

Twitter Bootstrap 3

Other Collapses: Collapses; bsCollapsePanel; updateCollapse

bsCollapsePanel *bsCollapsePanel*

Description

bsCollapsePanel creates individual panels within a bsCollapse object.

Usage

```
bsCollapsePanel(title, ..., value = title, style = NULL)
```

Arguments

title	The title to display at the top of the panel.
value	**Optional** The value to return when this panel is open. Defaults to title.
style	**Optional** A Bootstrap style to apply to the panel. (primary, danger, warning, info, or success)
...	UI elements to include within the panel.

Details

See Collapses for more information about how to use bsCollapsePanel with the rest of the Collapses family.

Note

Run bsExample("Collapses") for an example of bsCollapsePanel functionality.

See Also

Twitter Bootstrap 3

Other Collapses: Collapses; bsCollapse; updateCollapse

---

bsExample *bsExample*

Description

A function to view examples of shinyBS functionality. Will run the examples found in the examples sections of shinyBS documentation. Use this instead of example.

Usage

```
bsExample(family, display.mode = "showcase", ...)
```

Arguments

family	A shinyBS family name
display.mode	The display mode to use when running the example. See runApp.
...	Other parameters to pass to runApp.

Details

This function is just a wrapper for runApp that runs copies of the examples found in the family documention pages of shinyBS. By default, display.mode is set to showcase so you can see the code while the app is running.

Examples

```
## Not run:
   bsExample("Alerts")
## End(Not run)
```

---

bsModal *bsModal*

Description

bsModal is used within the UI to create a modal window.

Usage

```
bsModal(id, title, trigger, ..., size)
```

Arguments

id	A unique identifier for the modal window
title	The title to appear at the top of the modal
trigger	The id of a button or link that will open the modal.
size	**Optional** What size should the modal be? (small or large)
...	UI elements to include within the modal

Details

See Modals for more information about how to use bsModal with the rest of the Modals family.

Note

Run bsExample("Modals") for an example of bsModal functionality.

See Also

Twitter Bootstrap 3

Other Modals: Modals; toggleModal

bsPopover *bsPopover*

Description

bsPopover is used within the UI of an app to add a popover to a Shiny input or output.

Usage

```
bsPopover(id, title, content, placement = "bottom", trigger = "hover",
  options = NULL)
```

Arguments

id	The id of the element to attach the popover to.
title	The title of the popover.
content	The main content of the popover.
placement	Where the popover should appear relative to its target (top, bottom, left, or right). Defaults to "bottom".
trigger	What action should cause the popover to appear? (hover, focus, click, or manual). Defaults to "hover".
options	A named list of additional options to be set on the popover.

Details

See Tooltips_and_Popovers for more information about how to use bsPopover with the rest of the Tooltips_and_Popovers family.

Note

Run bsExample("Tooltips_and_Popovers") for an example of bsPopover functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: Tooltips_and_Popovers; addPopover; addTooltip; bsTooltip; popify; removePopover; removeTooltip; tipify

bsTooltip *bsTooltip*

Description

bsTooltip is used within the UI of an app to add a tooltip to a Shiny input or output.

Usage

```
bsTooltip(id, title, placement = "bottom", trigger = "hover",
  options = NULL)
```

Arguments

id	The id of the element to attach the tooltip to.
title	The content of the tooltip.
placement	Where the tooltip should appear relative to its target (top, bottom, left, or right). Defaults to "bottom".
trigger	What action should cause the tooltip to appear? (hover, focus, click, or manual). Defaults to "hover".
options	A named list of additional options to be set on the tooltip.

Details

See Tooltips_and_Popovers for more information about how to use bsTooltip with the rest of the Tooltips_and_Popovers family.

Note

Run bsExample("Tooltips_and_Popovers") for an example of bsTooltip functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: Tooltips_and_Popovers; addPopover; addTooltip; bsPopover; popify; removePopover; removeTooltip; tipify

Buttons *Buttons*

Description

Twitter Bootstrap gives many options for styling buttons that aren't made available by standard Shiny. Use shinyBS to create buttons of different sizes, shapes, and colors.

Details

Create a button in the UI with `bsButton`. If `type = "action"` the button will behave like the standard `actionButton` in shiny. If `type = "toggle"` the button will behave like a `checkboxInput` with an on and off state. It will return `TRUE` or `FALSE` to the Server depending on its state.

You can update the style and state of a `bsButton` from the Server logic with `updateButton`. For example, a button could be set to `disabled = TRUE` until the user has made some other selections, then once those selections have been made, an observer on the Server could use `updateButton` to enable the button allowing the user to proceed. Alternatively, you could set the button to `style = "success"` to let them know that the button is ready to be clicked.

Components

There are two functions in the Buttons family:

`bsButton` Used in the UI to create a button. Buttons can be of the type `action` or `toggle`.

`updateButton` Used in the Server logic to modify the state of a button created with `bsButton`

Changes

`bsActionButton` and `bsToggleButton` were replaced with just `bsButton` with a `type` argument.

`icon` was added to allow placing an icon in the button.

Note

Run `bsExample("Buttons")` for an example of `Buttons` functionality.

See Also

Twitter Bootstrap 3

Other Buttons: `bsButton`; `updateButton`

Examples

```
library(shiny)
library(shinyBS)
app = shinyApp(
 ui =
   fluidPage(
```

```
    sidebarLayout(
      sidebarPanel(
        sliderInput("bins",
                    "Move the slider to see its effect on the button below:",
                    min = 1,
                    max = 50,
                    value = 1),
        bsButton("actTwo", label = "Click me if you dare!", icon = icon("ban")),
     tags$p("Clicking the first button below changes the disabled state of the second button."),
        bsButton("togOne", label = "Toggle button disabled status",
                 block = TRUE, type = "toggle", value = TRUE),
        bsButton("actOne", label = "Block Action Button", block = TRUE)

      ),
      mainPanel(
        textOutput("exampleText")
      )
    )
  ),
server =
  function(input, output, session) {
    observeEvent(input$togOne, ({
      updateButton(session, "actOne", disabled = !input$togOne)
    }))
    observeEvent(input$bins, ({

      b <- input$bins
      disabled = NULL
      style = "default"
      icon = ""

      if(b < 5) {
        disabled = TRUE
        icon <- icon("ban")
      } else {
        disabled = FALSE
      }

      if(b < 15 | b > 35) {
        style = "danger"
      } else if(b < 20 | b > 30) {
        style = "warning"
      } else {
        style = "default"
        icon = icon("check")
      }

      updateButton(session, "actTwo", disabled = disabled, style = style, icon = icon)

    }))

    output$exampleText <- renderText({
      input$actTwo
```

```
        b <- isolate(input$bins)
        txt = ""
        if((b > 5 & b < 15) | b > 35) {
          txt = "That was dangerous."
        } else if((b > 5 & b < 20) | b > 30) {
          txt = "I warned you about that."
        } else if(b >= 20 &  b <= 30) {
          txt = "You have chosen... wisely."
        }
        return(txt)
      })
    }
)
## Not run:
 runApp(app)

## End(Not run)
```

`closeAlert` *closeAlert*

Description

`closeAlert` is used within your Server logic to close an alert that you created with `createAlert`.

Usage

```
closeAlert(session, alertId)
```

Arguments

`session`	The session object passed to function given to shinyServer.
`alertId`	The id of the alert to be dismissed.

Details

See Alerts for more information about how to use `closeAlert` with the rest of the Alerts family.

Note

Run `bsExample("Alerts")` for an example of `closeAlert` functionality.

See Also

Twitter Bootstrap 3

Other Alerts: `Alerts`; `bsAlert`; `createAlert`

`Collapses` *Collapses*

Description

Collapse panels allow you to reduce clutter in your Shiny app by making panels of information that open and close with a user's click. Any type of content can go in a collapse panel. Standard Bootstrap styling options are available.

Details

Collapses are designed to mimic `tabsetPanel` in their implementation. Start with `bsCollapse` to create a panel group, then fill it with panels using `bsCollapsePanel`.

`bsCollapse` acts as an input, so you can retrieve which panels are open from the input object passed to the function in `shinyServer`.

`updateCollapse` can be used within your server logic to open/close collapse panels or to change their style.

Components

`bsCollapse` A container for holder the individual panels created by `bsCollapsePanel`.

`bsCollapsePanel` Creates an individual Collapse Panel that resides within a `bsCollapse`.

`updateCollapse` Used within your server logic to open/close collapse panels or change their style.

Changes

`style` is a new option that wasn't available in previous versions of shinyBS.

Note

Run `bsExample("Collapses")` for an example of `Collapses` functionality.

See Also

Twitter Bootstrap 3

Other Collapses: `bsCollapsePanel`; `bsCollapse`; `updateCollapse`

Examples

```
library(shiny)
library(shinyBS)

app = shinyApp(
 ui =
 fluidPage(
   sidebarLayout(
    sidebarPanel(HTML("This button will open Panel 1 using <code>updateCollapse</code>."),
```

```
                  actionButton("p1Button", "Push Me!"),
                  selectInput("styleSelect", "Select style for Panel 1",
                   c("default", "primary", "danger", "warning", "info", "success"))
      ),
      mainPanel(
        bsCollapse(id = "collapseExample", open = "Panel 2",
                   bsCollapsePanel("Panel 1", "This is a panel with just text ",
                    "and has the default style. You can change the style in ",
                    "the sidebar.", style = "info"),
                   bsCollapsePanel("Panel 2", "This panel has a generic plot. ",
                  "and a 'success' style.", plotOutput("genericPlot"), style = "success")
        )
      )
    )
  ),
  server =
  function(input, output, session) {
    output$genericPlot <- renderPlot(plot(rnorm(100)))
    observeEvent(input$p1Button, ({
      updateCollapse(session, "collapseExample", open = "Panel 1")
    }))
    observeEvent(input$styleSelect, ({
     updateCollapse(session, "collapseExample", style = list("Panel 1" = input$styleSelect))
    }))
  }
)
## Not run:
 runApp(app)

## End(Not run)
```

---

createAlert *createAlert*

Description

createAlert is used within the Server logic of your Shiny app to display an alert to the user.

Usage

```
createAlert(session, anchorId, alertId = NULL, title = NULL,
  content = NULL, style = NULL, dismiss = TRUE, append = TRUE)
```

Arguments

session	The session object passed to function given to shinyServer.
anchorId	The unique identifier of the anchor where the alert should be displayed.
alertId	**Optional** A unique identifier for the Alert.
title	**Optional** A title for the Alert.

content	The main body of the Alert. HTML tags are allowed.
style	A bootstrap style to apply. Defaults to info.
dismiss	logical Should the Alert be user dismissable? Defaults to TRUE.
append	logical Should the Alert be appended below existing Alerts? Default to TRUE.

Details

See Alerts for more information about how to use createAlert with the rest of the Alerts family.

Note

Run bsExample("Alerts") for an example of createAlert functionality.

See Also

Twitter Bootstrap 3

Other Alerts: Alerts; bsAlert; closeAlert

Modals *Modals*

Description

Modal windows are similar to popups but are rendered within the original window. They can contain any combination of shiny inputs, shiny outputs, and html. Possible uses include extra controls that you don't want cluttering up the main app display or help pages to explain your apps operation.

Details

Use bsModal in your UI to create a modal window. It works like Collapses or tabPanel, any non-named arguments will be passed as content for the modal.

Create a button or link and assign its inputId as the trigger in bsModal.

Components

There are only two functions in the Modals family:

bsModal Used in the UI to create a modal window.

toggleModal Used in the Server logic to open or close a modal window programmatically.

Changes

There is now a toggle argument in toggleModal that allows you to specify whether you want the modal to open or close.

The size argument in bsModal allows you to specify the size of the modal window. Either small or large.

Note

Run `bsExample("Modals")` for an example of `Modals` functionality.

See Also

Twitter Bootstrap 3

Other Modals: `bsModal`; `toggleModal`

Examples

```
library(shiny)
library(shinyBS)

app = shinyApp(
 ui =
 fluidPage(
   sidebarLayout(
     sidebarPanel(
       sliderInput("bins",
                   "Number of bins:",
                   min = 1,
                   max = 50,
                   value = 30),
       actionButton("tabBut", "View Table")
     ),

     mainPanel(
       plotOutput("distPlot"),
       bsModal("modalExample", "Data Table", "tabBut", size = "large",
         dataTableOutput("distTable"))
     )
   )
 ),
 server =
 function(input, output, session) {

   output$distPlot <- renderPlot({

     x    <- faithful[, 2]
     bins <- seq(min(x), max(x), length.out = input$bins + 1)

     # draw the histogram with the specified number of bins
     hist(x, breaks = bins, col = 'darkgray', border = 'white')

   })

   output$distTable <- renderDataTable({

     x    <- faithful[, 2]
     bins <- seq(min(x), max(x), length.out = input$bins + 1)

     # draw the histogram with the specified number of bins
```

```
      tab <- hist(x, breaks = bins, plot = FALSE)
      tab$breaks <- sapply(seq(length(tab$breaks) - 1), function(i) {
        paste0(signif(tab$breaks[i], 3), "-", signif(tab$breaks[i+1], 3))
      })
      tab <- as.data.frame(do.call(cbind, tab))
      colnames(tab) <- c("Bins", "Counts", "Density")
      return(tab[, 1:3])

    }, options = list(pageLength=10))

  }
)
## Not run:
 runApp(app)

## End(Not run)
```

---

popify *popify*

Description

popify can be wrapped around any shiny UI element to add a popover to the wrapped element. This should be a safer way to add popovers to elements created with renderUI.

Usage

```
popify(el, title, content, placement = "bottom", trigger = "hover",
  options = NULL)
```

Arguments

el	A shiny UI element.
title	The title of the popover.
content	The main content of the popover.
placement	Where the popover should appear relative to its target (top, bottom, left, or right). Defaults to "bottom".
trigger	What action should cause the popover to appear? (hover, focus, click, or manual). Defaults to "hover".
options	A named list of additional options to be set on the popover.

Details

See Tooltips_and_Popovers for more information about how to use popify with the rest of the Tooltips_and_Popovers family.

Note

Run `bsExample("Tooltips_and_Popovers")` for an example of `popify` functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: `Tooltips_and_Popovers`; `addPopover`; `addTooltip`; `bsPopover`; `bsTooltip`; `removePopover`; `removeTooltip`; `tipify`

`removePopover` *removePopover*

Description

`removePopover` is used within the Server logic of an app to remove an existing popover from a Shiny input or output.

Usage

```
removePopover(session, id)
```

Arguments

`session`	The session object passed to function given to shinyServer.
`id`	The id of the element to remove the popover from.

Details

See Tooltips_and_Popovers for more information about how to use `removePopover` with the rest of the Tooltips_and_Popovers family.

Note

Run `bsExample("Tooltips_and_Popovers")` for an example of `removePopover` functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: `Tooltips_and_Popovers`; `addPopover`; `addTooltip`; `bsPopover`; `bsTooltip`; `popify`; `removeTooltip`; `tipify`

`removeTooltip` *removeTooltip*

Description

`removeTooltip` is used within the Server logic of an app to remove an existing tooltip from a Shiny input or output.

Usage

```
removeTooltip(session, id)
```

Arguments

`session`	The session object passed to function given to shinyServer.
`id`	The id of the element to remove the tooltip from.

Details

See Tooltips_and_Popovers for more information about how to use `removeTooltip` with the rest of the Tooltips_and_Popovers family.

Note

Run `bsExample("Tooltips_and_Popovers")` for an example of `removeTooltip` functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: `Tooltips_and_Popovers`; `addPopover`; `addTooltip`; `bsPopover`; `bsTooltip`; `popify`; `removePopover`; `tipify`

`tipify` *tipify*

Description

`tipify` can be wrapped around any shiny UI element to add a tooltip to the wrapped element. This should be a safer way to add tooltips to elements created with `renderUI`.

Usage

```
tipify(el, title, placement = "bottom", trigger = "hover", options = NULL)
```

Arguments

el	A shiny UI element.
title	The content of the tooltip.
placement	Where the tooltip should appear relative to its target (top, bottom, left, or right). Defaults to "bottom".
trigger	What action should cause the tooltip to appear? (hover, focus, click, or manual). Defaults to "hover".
options	A named list of additional options to be set on the tooltip.

Details

See Tooltips_and_Popovers for more information about how to use tipify with the rest of the Tooltips_and_Popovers family.

Note

Run bsExample("Tooltips_and_Popovers") for an example of tipify functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: Tooltips_and_Popovers; addPopover; addTooltip; bsPopover; bsTooltip; popify; removePopover; removeTooltip

---

toggleModal *toggleModal*

Description

toggleModal is used within your Server logic to open or close a modal window.

Usage

```
toggleModal(session, modalId, toggle = "toggle")
```

Arguments

session	The session object passed to function given to shinyServer.
modalId	The id of the modal window you want to open/close
toggle	Should the modal window open, close, or toggle?

Details

See Modals for more information about how to use toggleModal with the rest of the Modals family.

Note

Run `bsExample("Modals")` for an example of `toggleModal` functionality.

See Also

Twitter Bootstrap 3

Other Modals: `Modals`; `bsModal`

`Tooltips_and_Popovers` *Tooltips and Popovers*

Description

Tooltips and Popovers allow you to add additional information about controls or outputs without cluttering up your user interface. You can add a tooltip to a button that displays on hover and better explains what the button will do, or you could add a popover to an output providing further analysis of that output.

Details

You can create tooltips and popovers from either the UI script or within the Server logic. `bsTooltip` and `bsPopover` are used in the UI, and `addTooltip` and `addPopover` are used in the Server logic. `tipify` and `popify` can be used within the UI or from within a `renderUI` in the Server logic. They also have the added advantage of not requiring that the UI element have an ID attribute.

Components

There are eight functions in the Tooltips and Popovers family:

`bsTooltip` Used in the UI to add a tooltip to an element in your UI.

`bsPopover` Used in the UI to add a popover to an element in your UI.

`tipify` Wrap any UI element in `tipify` to add a tooltip to the wrapped element. Preferred for elemented created with `renderUI`.

`popify` Wrap any UI element in `popify` to add a popover to the wrapped element. Preferred for elements created with `renderUI`.

`addTooltip` Used in the Server logic to add a tooltip to an element in your UI.

`addPopover` Used in the Server logic to add a popover to an element in your UI.

`removeTooltip` Used in the Server logic to remove a tooltip from an element in your UI.

`removePopover` Used in the Server logic to remove a popover from an element in your UI.

Changes

An `options` argument has been added to the creation functions to allow advanced users more control over how the tooltips and popovers appear. See the Twitter Bootstrap 3 documentation for more details.

Note

Tooltips and Popovers cannot contain shiny inputs or outputs.

There must be at least one `shinyBS` component in the UI of your app in order for the necessary dependencies to be loaded. Because of this, `addTooltip` and `addPopover` will not work if they are the only shinyBS components in your app.

Tooltips and popovers may not work on some of the more complex shiny inputs or outputs. If you encounter a problem with tooltips or popovers not appearing please file a issue on the github page so I can fix it.

Run `bsExample("Tooltips_and_Popovers")` for an example of `Tooltips_and_Popovers` functionality.

See Also

Twitter Bootstrap 3

Other Tooltips_and_Popovers: `addPopover`; `addTooltip`; `bsPopover`; `bsTooltip`; `popify`; `removePopover`; `removeTooltip`; `tipify`

Examples

```
library(shiny)
library(shinyBS)

app = shinyApp(
 ui =
 fluidPage(
   sidebarLayout(
     sidebarPanel(
       sliderInput("bins",
                   "Number of bins:",
                   min = 1,
                   max = 50,
                   value = 30),
     bsTooltip("bins", "The wait times will be broken into this many equally spaced bins",
         "right", options = list(container = "body"))
     ),
     mainPanel(
       plotOutput("distPlot"),
       uiOutput("uiExample")
     )
   )
 ),
 server =
 function(input, output, session) {
   output$distPlot <- renderPlot({

     # generate bins based on input$bins from ui.R
     x    <- faithful[, 2]
     bins <- seq(min(x), max(x), length.out = input$bins + 1)

     # draw the histogram with the specified number of bins
```

```
    hist(x, breaks = bins, col = 'darkgray', border = 'white')

  })
  output$uiExample <- renderUI({
    tags$span(
      popify(bsButton("pointlessButton", "Button", style = "primary", size = "large"),
        "A Pointless Button",
        "This button is <b>pointless</b>. It does not do <em>anything</em>!"),
      tipify(bsButton("pB2", "Button", style = "inverse", size = "extra-small"),
        "This button is pointless too!")
    )
  })
  addPopover(session, "distPlot", "Data", content = paste0("<p>Waiting time between ",
    "eruptions and the duration of the eruption for the Old Faithful geyser ",
    "in Yellowstone National Park, Wyoming, USA.</p><p>Azzalini, A. and ",
    "Bowman, A. W. (1990). A look at some data on the Old Faithful geyser. ",
    "Applied Statistics 39, 357-365.</p>"), trigger = 'click')
 }
)
## Not run:
 runApp(app)

## End(Not run)
```

---

updateButton *updateButton*

Description

updateButton is used in your Server logic to update the style or state of a button.

Usage

```
updateButton(session, inputId, label = NULL, icon = NULL, value = NULL,
  style = NULL, size = NULL, block = NULL, disabled = NULL)
```

Arguments

session	The session object passed to function given to shinyServer.
inputId	Specifies the input slot that will be used to access the value.
label	The contents of the button or link–usually a text label, but you could also use any other HTML, like an image.
icon	An optional icon to appear on the button.
value	**logical** If type = "toggle", the initial value of the button.
style	A Bootstrap style to apply to the button. (default, primary, success, info, warning, or danger)
size	The size of the button (extra-small, small, default, or large)
block	**logical** Should the button take the full width of the parent element?
disabled	**logical** Should the button be disabled (un-clickable)?

Details

Because of the way it is coded, updateButton may work on buttons not created by bsButton such as submitButton.

See Buttons for more information about how to use updateButton with the rest of the Buttons family.

Note

Run bsExample("Buttons") for an example of updateButton functionality.

See Also

Twitter Bootstrap 3

Other Buttons: Buttons; bsButton

---

updateCollapse *updateCollapse*

Description

updateCollapse is used within the Server logic of your Shiny app to modify a Collapse after load.

Usage

```
updateCollapse(session, id, open = NULL, close = NULL, style = NULL)
```

Arguments

session	The session object passed to function given to shinyServer.
id	The id of the Collapse object you want to change.
open	A vector of value (or title if no value was provided) values identifying the panels you want to open.
close	A vector of value (or title if no value was provided) values identifying the panels you want to close.
style	A named list of Bootstrap styles (primary, danger, info, warning, success, or default). The names should correspond to the value (or title if no value was provided) of the bsCollapsePanel you want to change.

Details

See Collapses for more information about how to use updateCollapse with the rest of the Collapses family.

Note

Run bsExample("Collapses") for an example of updateCollapse functionality.

See Also

Twitter Bootstrap 3

Other Collapses: `Collapses`; `bsCollapsePanel`; `bsCollapse`

Index