Package 'shiny.semantic'

May 29, 2017

Type Package

Title Semantic UI Support for Shiny

Version 0.1.1

Description Creating a great user interface for your Shiny apps can be a hassle, especially if you want to work purely in R and don't want to use, for instance HTML templates. This package adds support for a powerful UI library Semantic UI - <http://semantic-ui.com/>. It also supports universal UI input binding that works with various DOM elements.

BugReports https://github.com/Appsilon/shiny.semantic/issues

Encoding UTF-8

LazyData TRUE

License MIT + file LICENSE

Imports utils, shiny (>= 0.12.1), htmltools (>= 0.2.6), htmlwidgets (>= 0.8), purrr (>= 0.2.2)

RoxygenNote 6.0.1

NeedsCompilation no

Author Filip Stachura [aut, cre]

Maintainer Filip Stachura <filip@appsilondatascience.com>

Repository CRAN

Date/Publication 2017-05-29 08:05:27 UTC

R topics documented:

---

dropdown *Create dropdown Semantic UI component*

Description

This creates a default dropdown using Semantic UI styles with Shiny input. Dropdown is already initialized and available under input[[name]].

Usage

```
dropdown(name, choices, choices_value = choices, default_text = "Select",
  value = NULL)
```

Arguments

name	Input name. Reactive value is available under input[[name]].
choices	All available options one can select from.
choices_value	What reactive value should be used for corresponding choice.
default_text	Text to be visible on dropdown when nothing is selected.
value	Pass value if you want to initialize selection for dropdown.

Examples

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

  library(shiny)
  library(shiny.semantic)
  ui <- function() {
      shinyUI(
        semanticPage(
          title = "Dropdown example",
          suppressDependencies("bootstrap"),
          uiOutput("dropdown"),
          p("Selected letter:"),
          textOutput("selected_letter")
       )
     )
  }
  server <- shinyServer(function(input, output) {
     output$dropdown <- renderUI({
         dropdown("simple_dropdown", LETTERS, value = "A")
     })
     output$selected_letter <- renderText(input[["simple_dropdown"]])
  })

  shinyApp(ui = ui(), server = server)
```

```
}
```

`getDeps` *Add dashboard dependencies to html*

Description

Internal function that adds dashboard dependencies to html.

Usage

```
getDeps()
```

Value

Content with appended dependencies.

`semanticPage` *Semantic UI page*

Description

This creates a Semantic page for use in a Shiny app.

Usage

```
semanticPage(..., title = "")
```

Arguments

`...`	Other arguments to be added as attributes of the main div tag wrapper (e.g. style, class etc.)
`title`	A title to display in the browser's title bar.

shiny_input *Create universal Shiny input binding*

Description

Universal binding for Shiny input on custom user interface. Using this function one can create various inputs ranging from text, numerical, date, dropdowns, etc. Value of this input is extracted via jQuery using $().val() function and default exposed as serialized JSON to the Shiny server. If you want to change type of exposed input value specify it via type param. Currently list of supported types is "JSON" (default) and "text".

Usage

```
shiny_input(input_id, shiny_ui, value = NULL, type = "JSON")
```

Arguments

input_id	String with name of this input. Access to this input within server code is normal with input[[input_id]].
shiny_ui	UI of HTML component presenting this input to the users. This UI should allow to extract its value with jQuery $().val() function.
value	An optional argument with value that should be set for this input. Can be used to store persisten input valus in dynamic UIs.
type	Type of input value (could be "JSON" or "text").

shiny_text_input *Create universal Shiny text input binding*

Description

Universal binding for Shiny text input on custom user interface. Value of this input is extracted via jQuery using $().val() function. This function is just a simple binding over shiny_input. Please take a look at shiny_input documentation for more information.

Usage

```
shiny_text_input(...)
```

Arguments

...	Possible arguments are the same as in shiny_input() method: input_id, shiny_ui, value. Type is already predefined as "text"

tabset *Create Semantic UI tabs*

Description

This creates tabs with content using Semantic UI styles.

Usage

```
tabset(tabs, id = generate_random_id("menu"),
  menu_class = "top attached tabular",
  tab_content_class = "bottom attached segment")
```

Arguments

tabs	A list of tabs. Each tab is a list of two elements - first element defines menu item, second element defines tab content.
id	Id of the menu element (default: randomly generated id)
menu_class	Class for the menu element (default: "top attached tabular")
tab_content_class	Class for the tab content (default: "bottom attached segment")

uiicon *Create Semantic UI icon tag*

Description

This creates an icon tag using Semantic UI styles.

Usage

```
uiicon(type = "", ...)
```

Arguments

type	A name of an icon. Look at http://semantic-ui.com/elements/icon.html for all possibilities.
...	Other arguments to be added as attributes of the tag (e.g. style, class etc.)

`uirender` *Render semanticui htmlwidget*

Description

htmlwidget that adds semanticui dependencies and renders in viewer or rmarkdown.

Usage

```
uirender(ui, width = NULL, height = NULL, elementId = NULL)
```

Arguments

`ui`	UI, which will be wraped in an htmlwidget.
`width`	Fixed width for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.
`height`	Fixed height for widget (in css units). The default is NULL, which results in intelligent automatic sizing based on the widget's container.
`elementId`	Use an explicit element ID for the widget (rather than an automatically generated one).

Index