Package ‘shinymaterial’

April 12, 2017

Type Package

Title Implement Google’s Material Design in shiny applications

Version 0.1.0

Author Eric Anderson

Maintainer Eric Anderson <eric.ray.anderson@gmail.com>

Description This package allows shiny developers to incorporate UI elements based on Google’s Material Design. This is accomplished by leveraging the library materialize css.

License MIT + file LICENSE

Imports shiny (>= 0.7.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

`material_card` *Create a card that will contain UI content*

Description

UI content can be placed in cards to organize items on a page.

Usage

```
material_card(title, ...)
```

Arguments

`title`	String. The title of the card
`...`	The UI elements to place in the card

Examples

```
material_card(
  title = "Example Card",
  shiny::tags$h1("Card Content")
)
```

`material_column` *Create a column to organize UI content*

Description

UI content can be placed in columns to organize items on a page.

Usage

```
material_column(..., width = 6, offset = 0)
```

Arguments

`...`	The UI elements to place in the column
`width`	Integer. The width of the column. The value should be between 1 and 12
`offset`	Integer. The offset to the left of the column. The value should be between 0 and 11

Examples

```
material_column(
  width = 4,
  shiny::tags$h1("Column Content")
)
```

`material_input` *Create a shinymaterial input*

Description

Build a shinymaterial input of any available type.

Usage

```
material_input(type, input_id, label, ...)
```

Arguments

`type`	String. The type of input to be created. See section "Input Types" for list of available types.
`input_id`	String. The input identifier used to access the value.
`label`	String. Display label for input.
`...`	Additional arguments for the input type.

Input Types

- **button**
  - `icon` *(String. The name of the icon. Visit* http://materializecss.com/icons.html *for a list of available icons.)*
- **checkbox**
- **dropdown**
  - `choices` *(Named vector. The list of option names and underyling values.)*
  - `selected` *(String. The initiali selected underyling value.)*
  - `multiple` *(Boolean. Can multiple items be selected?)*
- **floating-button**
  - `icon` *(String. The name of the icon. Visit* http://materializecss.com/icons.html *for a list of available icons.)*
- **number-box**
  - `min_value` *(Number. The minimum allowable value.)*
  - `max_value` *(Number. The maximum allowable value.)*
  - `initial_value` *(Number. The initial value.)*
- **password-box**
- **radio-button**
  - `choices` *(Named vector. The list of option names and underyling values.)*
- **slider**
  - `min_value` *(Number. The minimum allowable value.)*
  - `max_value` *(Number. The maximum allowable value.)*
  - `initial_value` *(Number. The initial value.)*
- **switch**
  - `off_label` *(String. The label for the 'off' portion of the switch.)*
  - `on_label` *(String. The label for the 'on' portion of the switch.)*
- **text-box**

Examples

```
##-- button --##
material_input(
  type = "button",
  input_id = "example_button",
  label = "Button",
  icon = "done"
)
```

```
##-- checkbox --##
material_input(
  type = "checkbox",
  input_id = "example_checkbox",
  label = "Checkbox"
)

##-- dropdown --##
material_input(
  type = "dropdown",
  input_id = "example_dropdown",
  label = "Dropdown",
  choices = c(
    "Chicken" = "c",
    "Steak" = "s",
    "Fish" = "f"
  ),
  selected = c("c"),
  multiple = FALSE
)

##-- floating-button --##
material_input(
  type = "floating-button",
  input_id = "example_floating_button",
  label = "Floating Button",
  icon = "done"
)

##-- number-box --##
material_input(
  type = "number-box",
  input_id = "example_number_box",
  label = "Number Box",
  min_value = 1,
  max_value = 10,
  initial_value = 2
)

##-- password-box --##
material_input(
  type = "password-box",
  input_id = "example_password_box",
  label = "Password Box"
)

##-- radio-button --##
material_input(
  type = "radio-button",
  input_id = "example_radio_button",
  label = "Radio Button",
  choices = c(
    "Cake" = "c",
    "Pie" = "p",
    "Brownie" = "b"
  )
)
```

```
##-- slider --##
material_input(
  type = "slider",
  input_id = "example_slider",
  label = "Slider",
  min_value = 1,
  max_value = 10,
  initial_value = 2
)

##-- switch --##
material_input(
  type = "switch",
  input_id = "example_switch",
  label = "Switch",
  off_label = "Off",
  on_label = "On"
)

##-- text-box --##
material_input(
  type = "text-box",
  input_id = "example_text_box",
  label = "Text Box"
)
```

---

material_page *Create a shinymaterial page*

Description

Build a shinymaterial page.

Usage

```
material_page(title, ...)
```

Arguments

title	String. The title of the page.
...	The UI elements to place in the page

Examples

```
material_page(
  title = "Example Title",
  shiny::tags$h1("Page Content")
)
```

`material_parallax` *Create a parallax image*

Description

Use this function to create a parallax effect in your application.

Usage

```
material_parallax(image_source)
```

Arguments

`image_source`	String. The image file name. Place the image in a folder labeled 'www' at the same level as the application (server.R & ui.R)

Examples

```
material_parallax(
  image_source = "example_image.jpg"
)
```

`material_row` *Create a row to organize UI content*

Description

UI content can be placed in rows to organize items on a page.

Usage

```
material_row(...)
```

Arguments

`...`	The UI elements to place in the row

Examples

```
material_row(
  shiny::tags$h1("Row Content")
)
```

`material_side_nav` *Create a side-nav that contains UI content*

Description

UI content can be placed in side-nav.

Usage

```
material_side_nav(..., fixed = FALSE)
```

Arguments

`...`	The UI elements to place in the side-nav
`fixed`	A boolean. Set to TRUE to keep side-nav open on large screens.

Examples

```
material_side_nav(
  fixed = FALSE,
  shiny::tags$h1("Side-nav Content")
)
```

`material_tabs` *Place UI content within a tab*

Description

Use this function to create tabs in your application.

Usage

```
material_tabs(tabs)
```

Arguments

`tabs`	Named Vector. The tab display names as well as the tab ids.

Examples

```
material_tabs(
  tabs = c(
    "Example Tab 1" = "example_tab_1",
    "Example Tab 2" = "example_tab_2"
  )
)
```

`material_tab_content` *Place UI content within a tab*

Description

Use this function to place UI content within a specific tab.

Usage

```
material_tab_content(tab_id, ...)
```

Arguments

`tab_id`	String. The tab id to place the content in
`...`	The UI elements to place in the tab

Examples

```
material_tab_content(
  tab_id = "example_tab_1",
  shiny::tags$h1("Tab Content")
)
```

Index