

Package ‘shinymaterial’

July 4, 2017

Type Package

Title Implement Material Design in Shiny Applications

Version 0.3.0

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

Description Allows shiny developers to incorporate UI elements based on Google's Material design. See <<https://material.io/guidelines/>> for more information.

URL <https://ericrayanderson.github.io/shinymaterial/>

License GPL-3 | file LICENSE

Imports shiny (>= 0.7.0)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

material_button	2
material_card	2
material_checkbox	3
material_column	3
material_depth	4
material_dropdown	4
material_floating_button	5
material_input	6
material_modal	6
material_number_box	7
material_page	8
material_parallax	9
material_password_box	9
material_radio_button	10
material_row	10
material_side_nav	11
material_slider	11
material_switch	12
material_tabs	13
material_tab_content	13
material_text_box	14

Index**15**

material_button	Create a shiny material button
-----------------	--------------------------------

Description

Build a shiny material button. The initial value is zero, and increases by one on each press.

Usage

```
material_button(input_id, label, icon = NULL, depth = NULL, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The button text.
icon	String. The name of the icon. Leave empty for no icon. Visit http://materializecss.com/icons.html for a list of available icons.
depth	Integer. The amount of depth of the button. The value should be between 0 and 5. Leave empty for the default depth.
color	String. The color of the button. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors.

Examples

```
material_button(
  input_id = "example_button",
  label = "Button",
  icon = "cloud",
  depth = 5,
  color = "blue lighten-2"
)
```

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, ..., depth = NULL)
```

Arguments

title	String. The title of the card
...	The UI elements to place in the card
depth	Integer. The amount of depth of the card. The value should be between 0 and 5. Leave empty for the default depth.

Examples

```
material_card(
  title = "Example Card",
  depth = 5,
  shiny::tags$h5("Card Content")
)
```

material_checkbox	Create a shinymaterial checkbox
-------------------	---------------------------------

Description

Build a shinymaterial checkbox. The value is a boolean (TRUE if checked, FALSE if not checked).

Usage

```
material_checkbox(input_id, label, initial_value = FALSE, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The checkbox label.
initial_value	Boolean. Is the checkbox initially checked?
color	String. The color of the check. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_checkbox(
  input_id = "example_checkbox",
  label = "Checkbox",
  initial_value = TRUE,
  color = "#ef5350"
)
```

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_depth	<i>Add depth to a UI element</i>
----------------	----------------------------------

Description

Give a UI element the perception of depth by creating a shadow.

Usage

```
material_depth(..., depth = 4)
```

Arguments

...	The UI elements to apply the depth.
depth	Integer. The amount of depth. The value should be between 0 and 5. A value of 0 can be used to remove depth from objects that have depth by default.

Examples

```
material_depth(
  depth = 5,
  material_card(title = "Example Depth")
)
```

material_dropdown	<i>Create a shiny material dropdown</i>
-------------------	---

Description

Build a shiny material dropdown.

Usage

```
material_dropdown(input_id, label, choices = NULL, selected = NULL,
  multiple = NULL, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The dropdown label.
choices	Named vector. The option names and underlying values.
selected	String. The initially selected underlying value.
multiple	Boolean. Can multiple items be selected?
color	String. The color of the dropdown choices. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_dropdown(
  input_id = "example_dropdown",
  label = "Drop down",
  choices = c(
    "Chicken" = "c",
    "Steak" = "s",
    "Fish" = "f"
  ),
  selected = c("c"),
  multiple = FALSE,
  color = "#ef5350"
)
```

```
material_floating_button
```

Create a shiny material floating button

Description

Build a shiny material floating button. The initial value is zero, and increases by one on each press.

Usage

```
material_floating_button(input_id, icon = NULL, depth = NULL,
  color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
icon	String. The name of the icon. Leave empty for no icon. Visit http://materializecss.com/icons.html for a list of available icons.
depth	Integer. The amount of depth of the floating button. The value should be between 0 and 5. Leave empty for the default depth.
color	String. The color of the floating button. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors.

Examples

```
material_floating_button(
  input_id = "example_floating_button",
  icon = "mode_edit",
  depth = 5,
  color = "red lighten-3"
)
```

material_input	Create a shinymaterial input
----------------	------------------------------

Description

Build a shinymaterial input.

Usage

```
material_input(type, ...)
```

Arguments

type	String. The type of input to be created.
...	Additional arguments for the input.

Examples

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

material_modal	Place UI content in a modal
----------------	-----------------------------

Description

Put any UI object inside of a modal. The modal will open when the button is pressed.

Usage

```
material_modal(modal_id, button_text, title, ..., button_icon = NULL,
  floating_button = FALSE, button_depth = NULL, button_color = NULL)
```

Arguments

modal_id	String. The ID for the modal. Must be unique per application.
button_text	String. The text displayed on the modal trigger button.
title	String. The title of the modal window.
...	The UI elements to place in the modal
button_icon	String. The name of the icon. Visit http://materializecss.com/icons.html for a list of available icons.
floating_button	Boolean. Should the modal trigger button be a floating button?
button_depth	Integer. The amount of depth of the button. The value should be between 0 and 5. Leave empty for the default depth.
button_color	String. The color of the button. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors.

Examples

```
material_modal(
  modal_id = "example_modal",
  button_text = "Modal",
  title = "Example Modal Title",
  button_color = "red lighten-3",
  shiny::tags$p("Modal Content")
)
```

material_number_box	<i>Create a shinymaterial number box</i>
---------------------	--

Description

Build a shinymaterial number box.

Usage

```
material_number_box(input_id, label, min_value, max_value, initial_value,
  color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The number box label.
min_value	Number. The minimum allowable value.
max_value	Number. The maximum allowable value.
initial_value	Number. The initial value.
color	String. The accent color of the number box. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_number_box(
  input_id = "example_number_box",
  label = "number box",
  min_value = 5,
  max_value = 15,
  initial_value = 10,
  color = "#ef5350"
)
```

material_page	<i>Create a shinymaterial page</i>
---------------	------------------------------------

Description

Build a shinymaterial page.

Usage

```
material_page(..., title = "", nav_bar_color = NULL,
  background_color = "grey lighten-4")
```

Arguments

...	The UI elements to place in the page.
title	String. The title of the page.
nav_bar_color	Color of the navigation bar. Leave blank for the default color. Visit http://materializecss.com/color.html for a list of available colors.
background_color	Page background color. Leave blank for the default color. Visit http://materializecss.com/color.html for a list of available colors.

Examples

```
material_page(
  title = "Example Title",
  nav_bar_color = "red lighten-2",
  background_color = "blue lighten-4",
  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_password_box	Create a shinymaterial password box
-----------------------	-------------------------------------

Description

Build a shinymaterial password box.

Usage

```
material_password_box(input_id, label, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The password box label.
color	String. The accent color of the password box. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_password_box(  
  input_id = "example_password_box",  
  label = "password box",  
  color = "#ef5350"  
)
```

material_radio_button	Create a shinymaterial radio button
-----------------------	-------------------------------------

Description

Build a shinymaterial radio button.

Usage

```
material_radio_button(input_id, label, choices, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The radio button label.
choices	Named vector. The option names and underlying values.
color	String. The color of the radio buttons. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_radio_button(  
  input_id = "example_radio_button",  
  label = "Radio Button",  
  choices = c(  
    "Cake" = "c",  
    "Pie" = "p",  
    "Brownie" = "b"  
  ),  
  color = "#ef5350"  
)
```

material_row	Create a row to organize UI content
--------------	-------------------------------------

Description

UI content can be placed in a row 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, image_source = NULL,
  background_color = NULL)
```

Arguments

...	The UI elements to place in the side-nav.
fixed	Boolean. Set to TRUE to keep side-nav open on large screens.
image_source	String. The background image file name. Place the image in a folder labeled 'www' at the same level as the application (server.R & ui.R).
background_color	Side-nav background color. Leave blank for the default color. Visit http://materializecss.com/color.html for a list of available colors.

Examples

```
material_side_nav(
  fixed = FALSE,
  image_source = "example_image.jpg",
  background_color = "blue lighten-4",
  shiny::tags$h1("Side-nav Content")
)
```

material_slider	Create a shinymaterial slider
-----------------	-------------------------------

Description

Build a shinymaterial slider.

Usage

```
material_slider(input_id, label, min_value, max_value, initial_value,
  color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The slider label.
min_value	Number. The minimum value on the slider.
max_value	Number. The maximum value on the slider.
initial_value	Number. The initial value of the slider.
color	String. The slider color. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_slider(
  input_id = "example_slider",
  label = "slider",
  min_value = 5,
  max_value = 15,
  initial_value = 10,
  color = "#ef5350"
)
```

material_switch	Create a shiny material switch
-----------------	--------------------------------

Description

Build a shiny material switch. The value is a boolean (TRUE if 'on', FALSE if 'off').

Usage

```
material_switch(input_id, label, off_label = "", on_label = "",
  initial_value = FALSE, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The switch label.
off_label	String. The label for the 'off' position of the switch.
on_label	String. The label for the 'on' position of the switch.
initial_value	Boolean. Is the switch initially on?
color	String. The color of the switch. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_switch(
  input_id = "example_switch",
  label = "Switch",
  off_label = "Off",
  on_label = "On",
  initial_value = TRUE,
  color = "#ef5350"
)
```

material_tabs	<i>Place UI content within a tab</i>
---------------	--------------------------------------

Description

Use this function to create tabs in your application.

Usage

```
material_tabs(tabs, color = NULL)
```

Arguments

tabs	Named vector. The tab display names and corresponding tab ids.
color	String. The accent color of the tabs. Leave blank for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>Tab color requires using word forms of colors (e.g. 'deep-purple'). Also, lighten or darken effects do not work on tab colors.</i>

Examples

```
material_tabs(
  tabs = c(
    "Example Tab 1" = "example_tab_1",
    "Example Tab 2" = "example_tab_2"
  ),
  color = "deep-purple"
)
```

material_tab_content	<i>Place UI content within a tab</i>
----------------------	--------------------------------------

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 in which to place the UI content.
...	The UI elements to place in the tab.

Examples

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

material_text_box	<i>Create a shinymaterial text box</i>
-------------------	--

Description

Build a shinymaterial text box.

Usage

```
material_text_box(input_id, label, color = NULL)
```

Arguments

input_id	String. The input identifier used to access the value.
label	String. The text box label.
color	String. The accent color of the text box. Leave empty for the default color. Visit http://materializecss.com/color.html for a list of available colors. <i>This input requires using color hex codes, rather than the word form. E.g., '#ef5350', rather than 'red lighten-1'.</i>

Examples

```
material_text_box(  
  input_id = "example_text_box",  
  label = "text box",  
  color = "#ef5350"  
)
```

Index

material_button, [2](#)
material_card, [2](#)
material_checkbox, [3](#)
material_column, [3](#)
material_depth, [4](#)
material_dropdown, [4](#)
material_floating_button, [5](#)
material_input, [6](#)
material_modal, [6](#)
material_number_box, [7](#)
material_page, [8](#)
material_parallax, [9](#)
material_password_box, [9](#)
material_radio_button, [10](#)
material_row, [10](#)
material_side_nav, [11](#)
material_slider, [11](#)
material_switch, [12](#)
material_tab_content, [13](#)
material_tabs, [13](#)
material_text_box, [14](#)