

Package ‘tabler’

October 31, 2025

Title Create Dashboards with 'Tabler' and 'Shiny'

Version 0.1.0

Description Provides functions to build interactive dashboards combining the 'Tabler UI Kit' with 'Shiny', making it easy to create professional-looking web applications. 'Tabler' is fully responsive and compatible with all modern browsers. Offers customizable layouts and components built with 'HTML5' and 'CSS3'. The underlying 'Tabler' (<<https://github.com/tabler/tabler>>) and 'Tabler Icons' (<<https://github.com/tabler/tabler-icons>>) were pre-built from source to eliminate the need for 'Node.js' and 'NPM' on package installation.

URL <https://github.com/pachadotdev/tabler>

BugReports <https://github.com/pachadotdev/tabler/issues>

Depends R (>= 3.5.0)

License Apache License (>= 2)

Imports htmltools

Suggests DT, testthat (>= 3.0.0)

RoxygenNote 7.3.3

Encoding UTF-8

LazyData true

Config/testthat.edition 3

NeedsCompilation no

Author Mauricio Vargas Sepulveda [aut, cre],
Tabler [ctb, cph] (Tabler Dashboard)

Maintainer Mauricio Vargas Sepulveda <m.vargas.sepulveda@gmail.com>

Repository CRAN

Date/Publication 2025-10-31 17:50:08 UTC

Contents

card	2
icons	4
menu_dropdown	5
navbar_menu	5
page	6
sidebar_brand	8

Index

9

card	<i>Create a Tabler Card</i>
------	-----------------------------

Description

- Build a modern card component with optional header, body, and footer
- Small helper to render the page pretitle element used by Tabler examples
- Small helper to render the page title element used by Tabler examples
- Create the full page header structure with pretitle and title
- Display a key metric or value prominently
- Display an icon from Tabler Icons or other icon libraries
- Display important messages to users
- Create interactive buttons with Tabler styling

Usage

```

card(..., title = NULL, footer = NULL, status = NULL, class = NULL)

pre_title(text)

title(text)

page_header(title_text, pretitle_text = NULL, ...)

value_box(value, title, icon = NULL, color = "primary", width = 3)

icon(name, library = "tabler", class = NULL)

alert(..., type = "info", dismissible = FALSE, title = NULL)

button(
  label,
  href = NULL,
  onclick = NULL,
  color = "primary",

```

```
size = "md",
outline = FALSE,
icon = NULL,
icon_right = FALSE,
disabled = FALSE,
block = FALSE,
pill = FALSE,
square = FALSE,
loading = FALSE,
class = NULL,
type = "button",
...
)
```

Arguments

...	Additional HTML attributes
title	Alert title (optional)
footer	Card footer content (optional)
status	Card color status: "primary", "secondary", "success", "warning", "danger", etc.
class	Additional CSS classes
text	The title text to display
title_text	The main title text
pretitle_text	The pretitle text (optional)
value	Main value to display
icon	Icon to include
color	Button color theme
width	Column width (1-12)
name	Icon name
library	Icon library: "tabler", "bootstrap", "feather"
type	HTML button type attribute
dismissible	Whether alert can be dismissed
label	Button text
href	URL to link to (creates an anchor tag instead of button)
onclick	JavaScript to execute on click
size	Button size: "sm", "md", "lg"
outline	Use outline style
icon_right	Position icon on the right side
disabled	Whether the button is disabled
block	Make button full-width (block-level)
pill	Use pill-shaped button style
square	Use square button style
loading	Show loading spinner

Value

- An HTML tag representing the card
- An HTML tag representing the value box
- An HTML tag representing the icon
- An HTML tag representing the alert
- An HTML tag representing the button

icons

Tabler Icons Data

Description

A dataset containing Tabler icon names and their categories.

Usage

icons

Format

A data frame with 6,019 rows and 3 variables:

- root** The root category of the icon (e.g., "arrow/arrows", "brand", "device", etc.)
- icon** The full icon name (e.g., "arrow-up", "arrow-down", "circle-arrow-up", "circle-arrow-up-filled", etc.)
- filled** Logical indicating whether the icon is a filled or not.

Source

Derived from the icons from <https://github.com/tabler/tabler-icons>

Examples

```
icons[grep1("paw", icons$icon), ]
```

menu_dropdown	<i>Create a Dropdown Menu Item</i>
---------------	------------------------------------

Description

Create a nav-item with a dropdown menu using columns like the example

Usage

```
menu_dropdown(text, icon = NULL, href = NULL, items = list())
```

Arguments

text	Title of the dropdown
icon	Icon name (optional)
href	Link for the dropdown toggle (optional)
items	A list of character vectors or tags representing dropdown links; each element can be a character vector of length 2: c(text, href)

Value

An HTML tag representing the dropdown nav item

navbar_menu	<i>Build a standard navbar/header</i>
-------------	---------------------------------------

Description

Create the top header used in boxed layout. This function accepts raw ‘li‘ items (created by ‘menu_item’) or a ‘sidebar_brand()‘ attached as ‘title‘ attribute on a ‘ul‘.

Usage

```
navbar_menu(..., brand = NULL, show_theme_button = FALSE)
```

Arguments

...	Additional nodes to include (typically ‘menu_item()‘ elements)
brand	Optional brand for the navbar; either a string (text title), ‘sidebar_brand(...)‘ or a named list with elements ‘text‘ and ‘img‘ (URL/path).
show_theme_button	Whether to show the theme toggle buttons (default: ‘FALSE‘).

Value

An HTML tag for the header

*page**Create a Tabler Dashboard Page*

Description

Main function to create a complete dashboard page with Tabler theme

Container for dashboard content

Helper to create a top navigation header. This replaces the previous top-navbar behavior when users passed a header-like component.

Footer for the dashboard

Container for navigation items in sidebar

Container for navigation items in horizontal layout

Individual navigation item for sidebar

Container for multiple tab panels in tabbed layouts

Individual tab panel content

Usage

```
page(
  title = NULL,
  navbar = NULL,
  body = NULL,
  footer = NULL,
  layout = "boxed",
  theme = "light",
  color = "blue",
  show_theme_button = FALSE
)
body(..., class = NULL)

topbar(title = NULL, brand_image = NULL, ...)
footer(left = NULL, right = NULL)
sidebar_menu(..., title = NULL)
horizontal_menu(...)

menu_item(text, tab_name = NULL, icon = NULL, href = NULL, badge = NULL)
tab_items(...)
tab_item(tab_name, ...)
```

Arguments

title	Optional brand for the sidebar; either a string (text title) or a named list with elements ‘text’ and ‘img’ (URL/path) to render a brand image and title. Example: ‘title = list(text = "My App", img = "logo.png")’.
navbar	Dashboard navbar/menu. Can be: - ‘sidebar_menu()’ for a vertical sidebar - ‘horizontal_menu()’ for a horizontal menu
body	Dashboard body content
footer	Dashboard footer (optional)
layout	Layout type: “boxed”
theme	Default theme: “light” (default) or “dark”.
color	Color theme (optional): “blue” (default), “azure”, “indigo”, “purple”, “pink”, “red”, “orange”, “yellow”, “lime”, “green”, “teal”, “cyan”.
show_theme_button	Whether to show the theme toggle buttons (default: ‘FALSE’).
...	Content for this tab
class	Additional CSS classes
brand_image	URL or path to brand image
left	Left-aligned content
right	Right-aligned content
text	Item text/label
tab_name	Unique identifier for the tab (must match menuItem tab_name)
icon	Icon name (optional)
href	Link URL (optional, alternative to tab_name)
badge	Badge text (optional)

Value

- HTML tag with dependencies attached
- An HTML tag representing the body
- An HTML tag representing the top header
- An HTML tag representing the footer
- An HTML tag representing the sidebar menu
- An HTML tag representing the horizontal menu
- An HTML tag representing the menu item
- An HTML tag representing the tab items container
- An HTML tag representing the tab item

Examples

```
ui <- page(
  title = "Combo Dashboard",
  layout = "combo",
  navbar = list(
    top = topbar(title = "My App"),
    side = sidebar_menu(
      menu_item("Dogs", icon = "dog"),
      menu_item("Cats", icon = "cat")
    )
  ),
  body = body("Welcome to Tabler!")
)

server <- function(input, output, session) {}

# shinyApp(ui, server)
```

sidebar_brand *Create a Sidebar Brand*

Description

Helper to build a brand structure for ‘sidebar_menu(title = ...)‘.

Usage

```
sidebar_brand(text = NULL, img = NULL, href = "#")
```

Arguments

<code>text</code>	Brand text/label
<code>img</code>	Optional image URL/path for brand logo
<code>href</code>	Optional link URL for the brand (defaults to "#")

Value

A named list suitable to pass to ‘sidebar_menu(title =)‘

Examples

```
sidebar_brand("My App", img = "logo.png")
```

Index

- * **datasets**
 - icons, [4](#)
- [alert \(card\)](#), [2](#)
- [body \(page\)](#), [6](#)
- [button \(card\)](#), [2](#)
- [card](#), [2](#)
- [footer \(page\)](#), [6](#)
- [horizontal_menu \(page\)](#), [6](#)
- [icon \(card\)](#), [2](#)
- icons, [4](#)
- [menu_dropdown](#), [5](#)
- [menu_item \(page\)](#), [6](#)
- [navbar_menu](#), [5](#)
- [page](#), [6](#)
- [page_header \(card\)](#), [2](#)
- [pre_title \(card\)](#), [2](#)
- [sidebar_brand](#), [8](#)
- [sidebar_menu \(page\)](#), [6](#)
- [tab_item \(page\)](#), [6](#)
- [tab_items \(page\)](#), [6](#)
- [title \(card\)](#), [2](#)
- [topbar \(page\)](#), [6](#)
- [value_box \(card\)](#), [2](#)