

Package ‘card.pro’

July 22, 2025

Type Package

Title Lightweight Modern & Responsive Card Component for 'shiny'

Version 2.3.0

Maintainer Obinna Obianom <idonshayo@gmail.com>

Description Responsive and modern 'HTML' card essentials for 'shiny' applications and dashboards. This novel card component in 'Bootstrap' provides a flexible and extensible content container with multiple variants and options for building robust 'R' based apps e.g for graph build or machine learning projects. The features rely on a combination of 'jQuery' <<https://jquery.com>> and 'CSS' styles to improve the card functionality.

License MIT + file LICENSE

URL <https://cardpro.rpkg.net>

BugReports <https://github.com/oobianom/card.pro/issues>

Depends R (> 3.6)

Imports shiny, quickcode, htmltools

Suggests rmarkdown, knitr, qpdf, testthat

Encoding UTF-8

VignetteBuilder knitr

Language en-US

LazyData false

RoxygenNote 7.2.3

Config/testthat/edition 3

NeedsCompilation no

Author Obinna Obianom [aut, cre]

Repository CRAN

Date/Publication 2025-01-15 17:30:02 UTC

Contents

add.JQuery	2
altPanel	3
card.pro	4
collapseEntry	7
empty.server	8
moveable	9
nav	9
primePanel	10
progressInput	11
row	12
tabEntry	12
template.loc	13
titlePanel2	13
use.cardpro	14
wrapper	15

Index	17
--------------	-----------

add.JQuery	<i>Attach jQuery to current document</i>
------------	--

Description

The current function allows inclusion of jQuery in an HTML output

Usage

```
add.JQuery(version = "3.5.1")
```

Arguments

version	jQuery version
---------	----------------

Value

script tag with jQuery link

Note

This function holds significance across the entire package, particularly in scenarios where a specific HTML output lacks the inclusion of the jQuery script required by other functions within this package. It serves as a crucial mechanism for ensuring that functionalities within the package remain operable even when the necessary jQuery script is absent from the HTML page by default knitting. By providing a fallback option, this function enhances the robustness and reliability of the package, allowing it to gracefully handle situations where jQuery is not explicitly included. Its role is pivotal in maintaining consistent behavior and functionality across diverse HTML environments, thereby facilitating seamless integration and usage of the package across various web development contexts.

Examples

```
add.JQuery()  
add.JQuery("3.7.0") # add a specific version
```

altPanel	<i>New sidebar panel to display content</i>
----------	---

Description

Customizable sidebar panel for inclusion of various UI elements

Usage

```
altPanel(..., width = 4, border = FALSE, shadow = FALSE)
```

Arguments

...	List of content
width	Width of the sidebar panel
border	Should border be declared for the panel
shadow	Should a shadow be added to the panel

Value

Creates an alternate container for displaying contents

Note

For more information on the features of the sidebar panel, look through the Github examples

Examples

```
altPanel("content 2")
```

 card.pro

Generate a flexible and extensible card container

Description

Widely used card component from Bootstrap with improvements to improve utility

Usage

```
card.pro(
  ...,
  title,
  collapsed = FALSE,
  width = 12,
  tabs = NULL,
  icon = NULL,
  xtra.header.content = NULL,
  footer = NULL,
  togglebtn = TRUE,
  editbtn = TRUE,
  expandbtn = TRUE,
  colorbtn = TRUE,
  removebtn = TRUE,
  sortable = TRUE,
  sidebar = NULL,
  shadow = TRUE,
  header.bg = c("white", "green", "greenDark", "greenLight", "purple", "magenta", "pink",
    "pinkDark", "blueLight", "teal", "blue", "blueDark", "darken", "yellow", "orange",
    "orangeDark", "red", "redLight"),
  alert.text = NULL,
  alert.type = c("warning", "info", "success", "danger"),
  collapsibleGroup = NULL
)
```

Arguments

...	The elements to include within the body of the card
title	The text to display in the header title
collapsed	If TRUE, the card is collapsed. The default is FALSE
width	Select a width from 1 to 12 to indicate the size of the card
tabs	optional. A list containing tabs items, see example
icon	Header icon e.g. shiny::icon('fire')
xtra.header.content	additional header content e.g. "Hi"
footer	Footer content if any

togglebtn	show toggle button
editbtn	show header title edit button
expandbtn	show card expand button
colorbtn	show header color chooser
removebtn	show card closer button
sortable	If FALSE, the card will not be draggable
sidebar	optional. include side bar content
shadow	logical. add shadow to the card exterior
header.bg	Header background color style
alert.text	Enter text for the alert portion. Leave as NULL to exclude the alert
alert.type	Indicate the type of alert to include, choices are "warning", "info", "success", "danger"
collapsibleGroup	optional. A list collapsible items, see example

Value

HTML code of the container with a class called card.pro that holds the items

Note

For more information on the features of the card, visit the examples section of the help documentation

Examples

```
# Example 1
if (interactive()) {
  library(shiny)
  library(card.pro)

  shiny::shinyApp(
    ui = fluidPage(
      titlePanel("Change customizations.
Its preserved across refresh!"),
      moveable(
        card.pro(
          "Lorem ipsum odor amet.",
          title = "Box 1",
          icon = icon("fire"),
          tabs = list(
            tabEntry(
              "Tab 1",
              textInput("nor1", "Enter name"),
              "Content for 1"
            ),
            tabEntry(
              "Tab 2",
```

```

        actionButton("nor1", "Try click"),
        "sample"
      )
    )
  ),
  card.pro(
    "Another something",
    title = "Box 2",
    width = 6,
    header.bg = "blue",
    alert.text = "Its an alert!",
    colorbtn = FALSE,
    editbtn = FALSE
  )
)
),
server = empty.server
)
}

# Example 2
if (interactive()) {
  library(shiny)
  library(card.pro)

  shiny::shinyApp(
    ui = fluidPage(
      titlePanel("Change order and refresh, changes saved"),
      wrapper(
        moveable(
          card.pro(
            "A plot for test",
            textInput("test6", "Plot title"),
            actionButton("test9", "Make plot"),
            title = "Box 3",
            width = 6,
            header.bg = "red",
            collapsed = TRUE,
            sortable = FALSE
          ),
        ),
        card.pro(
          "Box 4 content",
          "shinyappstore.com",
          title = "Box 1",
          width = 8,
          header.bg = "red",
          sidebar = div(
            "My settings",
            textInput("test2", "Input change"),
            actionButton("test3", "Click to save")
          ),
          footer = list(
            "Footnote: example",

```

```

        actionButton("test19", "Upload something")
    )
)
)
),
server = empty.server
)
}

```

collapseEntry

*Create a collapsible container panel item***Description**

Create a collapsible container panel item that is enclosed by a list

Usage

```

collapseEntry(
  ...,
  title,
  collapsed = FALSE,
  color.off = "darken",
  color.on = "red"
)

```

Arguments

...	content of the collapsible container
title	title of the collapsible container
collapsed	whether the panel is collapsed or not
color.off	color of collapsible icon when hover off
color.on	color of collapsible icon when hover on

Details

Get color choices using quickcode::

Value

An list containing the title and content of a collapsible container

Examples

```
if (interactive()) {  
  library(shiny)  
  library(card.pro)  
  card.pro(  
    title = "Sample collapsible",width = 4,  
    collapsibleGroup = list(  
      collapseEntry(title = "Summary 1", collapsed = TRUE, "Convallis aesus."),  
      collapseEntry(title = "Summary 2", "eiusmod tempor incididunt")  
    )  
  )  
}
```

empty.server

Empty server function

Description

For use in simple shiny apps not requiring server functions

Usage

```
empty.server(input, output, session)
```

Arguments

input	the input object
output	the output object
session	the session object

Value

Empty server function

Examples

```
if(interactive()){  
  library(shiny)  
  library(card.pro)  
  ui = fluidPage("Obi Obianom")  
  shinyApp(ui = ui, server = empty.server)  
}
```

moveable	<i>Generate a moveable grid</i>
----------	---------------------------------

Description

A grid that holds draggable items

Usage

```
moveable(...)
```

Arguments

... The elements to include within the body of the moveable grid

Value

HTML code of a container that allows items within it to be draggable

Note

For more information on the features of a sortable grid, visit the examples section of the help documentation

Examples

```
moveable(  
  htmltools::tags$div("A"),  
  htmltools::tags$div("B")  
)
```

nav	<i>Nav tag</i>
-----	----------------

Description

A nav tag for creating HTML navigations

Usage

```
nav(class, id = NULL, role = NULL, ...)
```

Arguments

class	The class of the navigation container
id	The identification of the navigation container
role	The character role of the container on the page
...	The content of the container

Value

HTML content of a container with type nav

Examples

```
nav('sample','id1','sample','some content')
```

primePanel	<i>Main panel to display content</i>
------------	--------------------------------------

Description

Customizable main panel for inclusion of various UI elements

Usage

```
primePanel(..., width = 8, border = FALSE, shadow = FALSE)
```

Arguments

...	List of content
width	Width of the main panel
border	Should border be declared for the panel
shadow	Should a shadow be added to the panel

Value

Creates a container for displaying contents

Note

For more information on the features of the main panel, look through the Github examples

Examples

```
primePanel("content 1")
```

progressInput *Create simple progress bar*

Description

Create a progress bar within card container

Usage

```
progressInput(  
  id,  
  label = "",  
  value = "35%",  
  outer.value = value,  
  color = color.choice,  
  vertical = FALSE,  
  size = c("m", "s", "l"),  
  striped = FALSE  
)
```

Arguments

id	id of the container
label	title of the progress bar
value	value of the progress bar in percent
outer.value	value shown next to title
color	color of the progress bar
vertical	if the progress bar should be vertical or horizontal
size	size of progress bar
striped	whether to show the progressed bar background as striped

Value

an HTML content to display a progress bar

Examples

```
if (interactive()) {  
  library(shiny)  
  library(card.pro)  
  
  # Without much modification of defaults  
  progressInput(id="id1")  
  
  # Set color and value  
  progressInput(id="id1", value = "90%", color = "green")  
}
```

row	<i>Generate a row div</i>
-----	---------------------------

Description

A simple row div

Usage

```
row(...)
```

Arguments

... The elements to include within the body of the row

Value

An HTML containing elements of a container with class row to be embedded in a page

Examples

```
row(shiny::div(width=12,"Hello nextGenShinyApps"))
```

tabEntry	<i>Create a tab panel item</i>
----------	--------------------------------

Description

Create a tab panel item that is enclosed by a list

Usage

```
tabEntry(title, ...)
```

Arguments

title title of the tab
... content of the tab

Value

An list containing the title and content of a tab

Examples

```
if (interactive()) {
  card.pro(
    title = "Sample tabs",
    tabs = list(
      tabEntry("Summary", "Convallis aesus."),
      tabEntry("Summary", "nextGenShinyApps.")
    )
  )
}
```

template.loc	<i>Template location full text</i>
--------------	------------------------------------

Description

Fetch the location of the scripts

Usage

```
template.loc(template = "bundle")
```

Arguments

template The type of template to fetch

Value

A path for the location of the package

Examples

```
template.loc('bundle')
```

titlePanel2	<i>Create a title and footer bar</i>
-------------	--------------------------------------

Description

Create an alternative title bar

Usage

```
titlePanel2(  
  title,  
  rightContent = NULL,  
  windowTitle = title,  
  text.col = "#2a2725",  
  bg.col = "#ffffff"  
)  
  
footerPanel(  
  title = shiny::HTML("&copy; 2024"),  
  rightContent = NULL,  
  bg.col = "#2a2725",  
  text.col = "#f5f5f5"  
)
```

Arguments

title	title of the tab
rightContent	content of the right
windowTitle	window title
text.col	text color
bg.col	background color

Value

An list containing the title and content of a header or footer

Examples

```
if (interactive()) {  
  titlePanel2("Main title", "Right content | About me")  
  titlePanel2("Main title", action)  
}  
  
if (interactive()) {  
  footerPanel("Main footer @ 2024. All rights reserved", "Contact us")  
}
```

use.cardpro

Include stylesheets and scripts

Description

Use the package scripts and stylesheets in a page

Usage

```
use.cardpro(
  theme = letters[1:5],
  jquery = FALSE,
  jqueryui = TRUE,
  fontawesome = FALSE,
  template = "bundle",
  fix.header = FALSE
)
```

Arguments

theme	The template type
jquery	option. logical. include jquery
jqueryui	option. logical. include jquery UI
fontawesome	option. logical. include fontawesome
template	template folder
fix.header	logical. fix header if titlePanel2() is used in UI

Value

A list of files to be inserted in the header of a page

Note

Theme options are "a", "b", "c", "d" or "e"

Examples

```
if(interactive()){
  use.cardpro()
  use.cardpro(theme="a",jqueryui = TRUE, fontawesome = FALSE)
}
```

 wrapper

A wrapper for panels

Description

Create a wrapper div for pannels

Usage

```
wrapper(
  ...,
  bg = c("default", "primary", "secondary", "warning", "info", "danger", "success")
)
```

Arguments

... div contents
bg background color of the wrapper

Value

a container for other containers

Examples

```
wrapper(altPanel("hello"), shiny::mainPanel("test"))  
wrapper(shiny::mainPanel("hello"), shiny::column(width = 2, "test"))
```


Index

`add.JQuery`, 2
`altPanel`, 3

`card.pro`, 4
`collapseEntry`, 7

`empty.server`, 8

`footerPanel (titlePanel2)`, 13

`moveable`, 9

`nav`, 9

`primePanel`, 10
`progressInput`, 11

`row`, 12

`tabEntry`, 12
`template.loc`, 13
`titlePanel2`, 13

`use.cardpro`, 14

`wrapper`, 15