

Package ‘miniUI’

May 8, 2026

Type Package

Title Shiny UI Widgets for Small Screens

Version 0.1.2

Description Provides UI widget and layout functions for writing Shiny apps that work well on small screens.

License GPL-3

URL <https://github.com/rstudio/miniUI>

BugReports <https://github.com/rstudio/miniUI/issues>

Imports shiny (>= 0.13), htmltools (>= 0.3), utils

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Joe Cheng [cre, aut],
Posit Software, PBC [cph, fnd] (ROR: <<https://ror.org/03wc8by49>>)

Maintainer Joe Cheng <joe@posit.co>

Repository CRAN

Date/Publication 2025-04-17 23:20:02 UTC

Contents

miniButtonBlock	2
miniContentPanel	2
miniPage	3
miniTabstripPanel	4
miniTitleBar	5

Index	7
--------------	----------

miniButtonBlock *Create a block-level button container*

Description

Creates a full-width container for one or more buttons. The horizontal space will be evenly divided among any buttons that are added.

Usage

```
miniButtonBlock(..., border = "top")
```

Arguments

...	One or more <code>shiny::actionButton()</code> or <code>shiny::downloadButton()</code> objects.
border	Zero or more of c("top", "bottom"), indicating which sides should have borders, if any.

Details

When using `miniButtonBlock` with a `miniTabstripPanel`, consider passing the `miniButtonBlock` to `miniTabstripPanel` as the between argument.

See Also

For more information, see the [Designing Gadget UI](#) article on shiny.rstudio.com.

Examples

```
library(shiny)

miniButtonBlock(
  actionButton("reset", "Reset to defaults"),
  actionButton("clear", "Clear all")
)
```

miniContentPanel *Create a content panel*

Description

Creates a panel for containing arbitrary content within a flex box container. This is mainly useful within `miniPage()` or a `miniTabPanel()`. You can use `miniContentPanel` to introduce padding and/or scrolling, but even if padding/scrolling aren't needed, it's a good idea to wrap your custom content into `miniContentPanel` as it fixes some odd behavior with percentage-based heights.

Usage

```
miniContentPanel(..., padding = 15, scrollable = TRUE)
```

Arguments

...	UI objects to be contained in the miniContentPanel. A single htmlwidget or <code>shiny::plotOutput()</code> with <code>height="100%"</code> works well, as do <code>shiny::fillRow()/shiny::fillCol()</code> .
padding	Amount of padding to apply. Can be numeric (in pixels) or character (e.g. "3em").
scrollable	If TRUE, then content large enough to overflow the miniContentPanel will make scrollbars appear.

See Also

For more information, see the [Designing Gadget UI](#) article on shiny.rstudio.com.

Examples

```
library(shiny)

miniContentPanel(padding = 0,
  plotOutput("plot", height = "100%")
)
```

 miniPage

Page function for Shiny Gadgets

Description

Designed to serve as the outermost function call for your gadget UI. Similar to `shiny::fillPage()`, but always includes the Bootstrap CSS library, and is designed to contain `miniTitleBar()`, `miniTabstripPanel()`, `miniContentPanel()`, etc.

Usage

```
miniPage(..., title = NULL, theme = NULL)
```

Arguments

...	Elements to include within the page.
title	The title to use for the browser window/tab (it will not be shown in the document).
theme	URL to alternative Bootstrap stylesheet.

See Also

For more information, see the [Designing Gadget UI](#) article on shiny.rstudio.com.

```
miniTabstripPanel      Create a tabstrip panel
```

Description

miniTabstripPanel is a tabstrip panel that contains miniTabPanel elements. Similar to `shiny::tabsetPanel()`, but optimized for small page sizes like mobile devices or the RStudio Viewer pane.

Usage

```
miniTabstripPanel(..., id = NULL, selected = NULL, between = NULL)
```

```
miniTabPanel(title, ..., value = title, icon = NULL)
```

Arguments

...	For miniTabstripPanel, miniTabPanel elements to include in the tabset. For miniTabPanel, UI elements to include within the tab.
id	If provided, you can use <code>input\$id</code> in your server logic to determine which of the current tabs is active. The value will correspond to the value argument that is passed to <code>miniTabPanel()</code> .
selected	The value (or, if none was supplied, the title) of the tab that should be selected by default. If NULL, the first tab will be selected.
between	A tag or list of tags that should be inserted between the content (above) and tabstrip (below).
title	Display title for tab.
value	The value that should be sent when miniTabstripPanel reports that this tab is selected. If omitted and miniTabstripPanel has an id, then the tab's title will be used as the value.
icon	Icon to appear on the tab; see <code>shiny::icon()</code> .

See Also

For more information, see the [Designing Gadget UI](#) article on shiny.rstudio.com.

Examples

```
library(shiny)

miniTabstripPanel(
  miniTabPanel("Data", icon = icon("table"),
    selectInput("dataset", "Data set", ls("package:datssets"))),
  miniTabPanel("Subset", icon = icon("sliders"),
    uiOutput("subset_ui")
  )
)
```

miniTitleBar	<i>Create a title bar</i>
--------------	---------------------------

Description

Creates a title bar for a Shiny app or Shiny Gadget. Intended to be used with `miniPage()`. Title bars contain a title, and optionally, a `miniTitleBarButton` on the left and/or right sides.

Usage

```
miniTitleBar(title, left = NULL, right = NULL)

gadgetTitleBar(
  title,
  left = miniTitleBarCancelButton(),
  right = miniTitleBarButton("done", "Done", primary = TRUE)
)

miniTitleBarButton(inputId, label, primary = FALSE)

miniTitleBarCancelButton(inputId = "cancel", label = "Cancel", primary = FALSE)
```

Arguments

title	The title of the gadget. If this needs to be dynamic, pass <code>shiny::textOutput()</code> with <code>inline = TRUE</code> .
left	The <code>miniTitleBarButton</code> to put on the left, or <code>NULL</code> for none.
right	The <code>miniTitleBarButton</code> to put on the right, or <code>NULL</code> for none.
inputId	The input slot that will be used to access the button.
label	The text label to display on the button.
primary	If <code>TRUE</code> , render the button in a bold color to indicate that it is the primary action of the gadget.

Details

`gadgetTitleBar` is a `miniTitleBar` with different defaults: a Cancel button on the left and a Done button on the right. By default, `shiny::runGadget()` will handle the Cancel button by closing the gadget and raising an error, but the Done button must be handled by the gadget author using `observeEvent(input$done, {...})`.

`miniTitleBarCancelButton` is like `miniTitleBarButton`, but the user can also invoke it by hitting the Escape key.

See Also

For more information, see the [Designing Gadget UI](#) article on shiny.rstudio.com.

Examples

```
miniTitleBar("My App",  
  left = miniTitleBarButton("prev", "Previous"),  
  right = miniTitleBarButton("next", "Next")  
)
```

Index

gadgetTitleBar (miniTitleBar), 5

miniButtonBlock, 2
miniContentPanel, 2
miniContentPanel(), 3
miniPage, 3
miniPage(), 2, 5
miniTabPanel (miniTabstripPanel), 4
miniTabPanel(), 2, 4
miniTabstripPanel, 4
miniTabstripPanel(), 3
miniTitleBar, 5
miniTitleBar(), 3
miniTitleBarButton (miniTitleBar), 5
miniTitleBarCancelButton
 (miniTitleBar), 5

shiny:::actionButton(), 2
shiny:::downloadButton(), 2
shiny:::fillCol(), 3
shiny:::fillPage(), 3
shiny:::fillRow(), 3
shiny:::icon(), 4
shiny:::plotOutput(), 3
shiny:::runGadget(), 5
shiny:::tabsetPanel(), 4
shiny:::textOutput(), 5