

Package ‘shiny.emptystate’

May 9, 2026

Title Empty State Components for 'Shiny'

Version 0.1.0

Description Offers a comprehensive solution for managing 'empty states' in 'Shiny' applications. It provides tools to create both default and customizable components for scenarios where data is absent or doesn't match user-defined filters. The package prioritizes user experience, ensuring clarity and consistency even when data is not available to display.

URL <https://appsilon.github.io/shiny.emptystate/>,
<https://github.com/Appsilon/shiny.emptystate>

BugReports <https://github.com/Appsilon/shiny.emptystate/issues>

License LGPL-3

Imports fontawesome, htmltools, R6, shiny, utils

Suggests bsicons, chromote, covr, knitr, lintr, rcmdcheck, rmarkdown,
shinytest2, spelling, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Ryszard Szymański [aut, cre],
Reclé Vibal [aut],
Harsh Verma [ctb],
Vedha Viyash [ctb]

Maintainer Ryszard Szymański <opensource+ryszard@appsilon.com>

Repository CRAN

Date/Publication 2023-08-17 06:50:06 UTC

Contents

default_empty_state_component	2
EmptyStateManager	2
empty_state_component	4
use_empty_state	5

Index[7](#)

default_empty_state_component
Default empty state component

Description

Default empty state component, used when user doesn't provide any value while initializing new EmptyStateManager object.

Usage

default_empty_state_component()

Value

a shiny.tag.

EmptyStateManager *EmptyStateManager R6 class*

Description

Creates an EmptyStateManager to then show or hide content.

Details

Creates an object to show an empty state content on selected id specified by id parameter. Then show or hide or use is_empty_state_show to check the status.

Value

EmptyStateManager R6 class

Methods**Public methods:**

- [EmptyStateManager\\$new\(\)](#)
- [EmptyStateManager\\$is_empty_state_show\(\)](#)
- [EmptyStateManager\\$show\(\)](#)
- [EmptyStateManager\\$hide\(\)](#)
- [EmptyStateManager\\$clone\(\)](#)

Method new(): Creates a new empty state manager object.

Usage:

```
EmptyStateManager$new(  
  id,  
  html_content = default_empty_state_component(),  
  color = NULL  
)
```

Arguments:

`id` id of element which should be covered with `html_content`
`html_content` Content for empty state. Defaults to `default_empty_state_component()`
`color` Background color of empty state content. Defaults to `NULL`

Returns: A new `EmptyStateManager` R6 class object.

Method `is_empty_state_show()`: Returns the current visibility state of the empty state UI. Defaults to `FALSE`

Usage:

```
EmptyStateManager$is_empty_state_show()
```

Returns: boolean, `TRUE/FALSE`

Method `show()`: Show empty state component.

Usage:

```
EmptyStateManager$show()
```

Returns: Nothing, it changes state of empty state

Method `hide()`: Hides empty state component.

Usage:

```
EmptyStateManager$hide()
```

Returns: Nothing, it changes state of empty state

Method `clone()`: The objects of this class are cloneable with this method.

Usage:

```
EmptyStateManager$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Examples

```
library(shiny)  
library(shiny.emptystate)  
library(fontawesome)  
  
ui <- fluidPage(  
  use_empty_state(),  
  actionButton("show", "Show empty state!"),  
  actionButton("hide", "Hide empty state!"),  
  tableOutput("my_table")  
)
```

```
server <- function(input, output) {  
  # Creating a custom empty state component  
  empty_state_content <- empty_state_component(  
    content = fa("eye-slash", height = "10rem", fill = "#808000"),  
    title = "Hide empty state to see table",  
    subtitle = "This empty state uses a FontAwesome icon."  
  )  
  
  # Initialize a new empty state manager object  
  manager_object <- EmptyStateManager$new(  
    id = "my_table",  
    html_content = empty_state_content  
  )  
  
  observeEvent(input$show, {  
    # Show empty state  
    manager_object$show()  
  })  
  
  observeEvent(input$hide, {  
    # Hide empty state  
    manager_object$hide()  
  })  
  
  output$my_table <- renderTable(mtcars)  
}  
  
if (interactive()) {  
  shinyApp(ui = ui, server = server)  
}
```

empty_state_component *Create an empty state component*

Description

Function to create a custom empty state component.

Usage

```
empty_state_component(content, title, subtitle = NULL)
```

Arguments

content	An HTML tag object used to render and provide the main content for the empty state.
title	A character string representing the main text that describes the empty state content.
subtitle	A character string providing supporting details about the empty state. Defaults to NULL

Details

content works best with `fontawesome::fa()` and `bsicons::bs_icon()`. `shiny::icon()` will also work, but this will require loading the html dependencies in the ui, i.e. calling `fontawesome::fa_html_dependency()` to use icons from FontAwesome. Glyphicon does not need any html dependency.

Value

a shiny.tag

Examples

```
library(shiny.emptystate)

if (interactive()) {
  empty_state_component(
    fontawesome::fa(name = "clipboard-question", height = "10rem"),
    title = "Content is not available",
    subtitle = "Please provide valid inputs to generate content."
  )

  empty_state_component(
    bsicons::bs_icon(
      name = "question-square",
      size = "15rem"
    ),
    title = "Content is not available",
    subtitle = "Please provide valid inputs to generate content."
  )
}
```

use_empty_state

Add empty state dependency

Description

Empty state dependencies to include anywhere in your UI but ideally at the top.

Usage

```
use_empty_state()
```

Value

a `html_dependency` object

Examples

```
library(shiny)
library(shiny.emptystate)

if (interactive()) {
  ui <- fluidPage(
    use_empty_state(),
    dataTableOutput("my_table")
  )
}
```

Index

`bsicons::bs_icon()`, 5

`default_empty_state_component`, 2

`empty_state_component`, 4

`EmptyStateManager`, 2

`fontawesome::fa()`, 5

`fontawesome::fa_html_dependency()`, 5

`shiny::icon()`, 5

`use_empty_state`, 5