

Package ‘cascadeSelect’

May 8, 2026

Title A Cascade Select Input for 'Shiny'

Version 1.1.0

Description Provides a cascade select widget for usage in 'Shiny' applications. This is useful for selection of hierarchical choices (e.g. continent, country, city). It is taken from the 'JavaScript' library 'PrimeReact'.

License GPL-3

Depends R (>= 2.10)

Imports fontawesome, grDevices, htmltools, reactR, shiny, tools

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

URL <https://github.com/stla/cascadeSelect>

BugReports <https://github.com/stla/cascadeSelect/issues>

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
PrimeTek Informatique [cph] (PrimeReact library)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2023-06-15 06:00:02 UTC

Contents

cascadeSelectInput	2
Icon	6
oilcons	7
pi_icons	7
themes	8

Index	9
--------------	----------

cascadeSelectInput *Cascade select input*

Description

Create a cascade select input for Shiny.

Usage

```
cascadeSelectInput(  
  inputId,  
  choices,  
  selected = NULL,  
  placeholder = "Select",  
  optionLabel,  
  optionGroupLabel,  
  optionGroupChildren,  
  theme = "bootstrap4-dark-purple"  
)
```

Arguments

inputId	the id that will be used to get the selected value in Shiny
choices	a hierarchical list (see the example); each item is given by a list with <i>must</i> contain an icon field created with the Icon function
selected	the selected value; NULL for none
placeholder	placeholder appearing when no selected value
optionLabel	the label of the options to be selected
optionGroupLabel	the label of the groups of options; there can be several groups and they must have the same label
optionGroupChildren	a list of the names of the groups of options
theme	the CSS theme; see <code>data(themes)</code> for the list of available themes

Value

A `shiny.tag.list` object to be included in a Shiny UI.

Examples

```
library(shiny)  
library(cascadeSelect)  
  
## | the hierarchical list of choices  
folders <- list()
```

```
list( # first folder
  name = "bootstrap", icon = Icon("bi bi-bootstrap", color = "purple"),
  subfolders = list(
    list( # subfolder of the first folder
      name = "css", icon = Icon("bi bi-folder-fill", color = "orange"),
      files = list(
        list(
          fname = "bootstrap-theme.css", size = "25 KB",
          icon = Icon("bi bi-filetype-css", color = "steelblue")
        ),
        list(
          fname = "bootstrap.css", size = "142 KB",
          icon = Icon("bi bi-filetype-css", color = "steelblue")
        )
      )
    ),
    list( # subfolder of the first folder
      name = "js", icon = Icon("bi bi-folder-fill", color = "orange"),
      files = list(
        list(
          fname = "bootstrap.js", size = "74 KB",
          icon = Icon("bi bi-filetype-js", color = "yellow")
        ),
        list(
          fname = "npm.js", size = "484 B",
          icon = Icon("bi bi-filetype-js", color = "yellow")
        )
      )
    )
  ),
list( # second folder
  name = "datatables", icon = Icon("bi bi-table", color = "purple"),
  subfolders = list(
    list( # subfolder of the second folder
      name = "css", icon = Icon("bi bi-folder-fill", color = "orange"),
      files = list(
        list(
          fname = "dataTables.bootstrap.css", size = "7.5 KB",
          icon = Icon("bi bi-filetype-css", color = "steelblue")
        ),
        list(
          fname = "dataTables.extra.css", size = "1.2 KB",
          icon = Icon("bi bi-filetype-css", color = "steelblue")
        )
      )
    ),
    list( # subfolder of the second folder
      name = "js", icon = Icon("bi bi-folder-fill", color = "orange"),
      files = list(
        list(
          fname = "dataTables.bootstrap.js", size = "4.2 KB",
          icon = Icon("bi bi-filetype-js", color = "yellow")
```

```

    ),
    list(
      fname = "jquerydataTable.min.js", size = "77.1 KB",
      icon = Icon("bi bi-filetype-js", color = "yellow")
    )
  )
)
)
)
)
)

## | the Shiny app
ui <- fluidPage(
  titlePanel("Cascade Select"),
  fluidRow(
    column(
      6,
      cascadeSelectInput(
        "cascade",
        choices = folders,
        placeholder = "Select a file",
        optionLabel = "fname",
        optionGroupLabel = "name",
        optionGroupChildren = list("subfolders", "files"),
        theme = "bootstrap4-dark-purple"
      ),
      br(),br(),
      uiOutput("textOutput")
    )
  )
)

server <- function(input, output, session) {
  output[["textOutput"]] <- renderUI({
    choice <- req(input[["cascade"]])
    tagList(
      tags$h4("You selected the file: ", sQuote(choice[["fname"]]), "."),
      tags$h4("Its size is: " , choice[["size"]], ".")
    )
  })
}

if(interactive()) {
  shinyApp(ui, server)
}

# other example, with different group depths ###
library(shiny)
library(cascadeSelect)

folderHaskell <- list(
  list( # first folder

```

```

name = "findPatternInFiles",
icon = Icon("bi bi-folder-fill", color = "orange"),
sub = list(
  list( # subfolder of the first folder
    name = "src", icon = Icon("bi bi-folder-fill", color = "orange"),
    subsub = list(
      list( # file
        fname = "GetAhaHTML.hs", icon = Icon("oi oi-haskell")
      ),
      list( # file
        fname = "GetGrepResults.hs", icon = Icon("oi oi-haskell")
      )
    )
  ),
  list( # subfolder of the first folder
    name = "src-exe", icon = Icon("bi bi-folder-fill", color = "orange"),
    subsub = list(
      list( # file
        fname = "Main.hs", icon = Icon("oi oi-haskell")
      )
    )
  ),
  list( # file in the first folder
    fname = "findPatternInFiles.cabal", icon = Icon("oi oi-cabal")
  ),
  list( # file in the first folder
    fname = "LICENSE", icon = Icon("oi oi-license")
  ),
  list( # file in the first folder
    fname = "README.md", icon = Icon("oi oi-markdown")
  ),
  list( # file in the first folder
    fname = "Setup.hs", icon = Icon("oi oi-haskell")
  ),
  list( # file in the first folder
    fname = "stack.yaml", icon = Icon("oi oi-yaml")
  ),
  list( # file in the first folder
    fname = ".gitignore", icon = Icon("bi bi-git")
  )
)
)
)

ui <- fluidPage(
  titlePanel("My Haskell project"),
  fluidRow(
    column(
      6,
      cascadeSelectInput(
        "cascade",
        choices = folderHaskell,
        placeholder = "Select a file",

```

```

        optionLabel = "fname",
        optionGroupLabel = "name",
        optionGroupChildren = list("sub", "subsub"),
        theme = "luna-amber"
      ),
      br(),br(),
      textOutput("textOutput")
    )
  )
)

server <- function(input, output, session) {
  output[["textOutput"]] <- renderText({
    choice <- input[["cascade"]]
    sprintf(
      "You selected the file: %s.", dQuote(choice[["fname"]])
    )
  })
}

if(interactive()) {
  shinyApp(ui, server)
}

```

 Icon

Create an icon

Description

Create an icon with a given color and a given size.

Usage

```
Icon(icon = "pi pi-circle-fill", color = "red", size = "1.5rem", scale = 1)
```

Arguments

icon	the name (actually the class name) of the icon; PrimeIcons are available, for example "pi pi-youtube" (see PrimeIcons), fontawesome icons are available, for example "fa fa-paper-plane", and Bootstrap icons are available, for example "bi bi-apple" (see Bootstrap icons)
color	a CSS color, e.g. "crimson" or a hex code like "#ffaa07"
size	size of the icon, a css measurement (e.g. 1rem, 11px); for the oi family of icons (see oiIcons), do not change the default size and use the scale argument instead
scale	a positive number, the scale for an oi icon (see oiIcons); you must use this argument to control its size, not the size argument

Value

A list to be included in the fields icon of the choices list of [cascadeSelectInput](#).

Note

The color argument has no effect on the oi icons.

oiIcons	<i>The "oi" icons</i>
---------	-----------------------

Description

The oi icons are some SVG icons. This function lists them. They are intended to be used in the [Icon](#) function.

Usage

```
oiIcons()
```

Value

A data frame.

pi_icons	<i>The prime icons</i>
----------	------------------------

Description

The names of the icons available in the **PrimeIcons** library.

Usage

```
pi_icons
```

Format

A vector with 260 elements. Each element is the name of an icon. In order to use it in the [Icon](#) function, you have to prefix it with pi pi- (for example "pi pi-youtube"). See [PrimeIcons](#) for the list of all icons.

themes

The CSS themes

Description

The names of the CSS themes available in the **PrimeReact** library.

Usage

themes

Format

A vector with 34 elements. Each element is the name of a theme, to be used as the theme argument of the [cascadeSelectInput](#) function.

Index

* datasets

pi_icons, [7](#)

themes, [8](#)

[cascadeSelectInput](#), [2](#), [7](#), [8](#)

[Icon](#), [2](#), [6](#), [7](#)

[oiIcons](#), [6](#), [7](#)

[pi_icons](#), [7](#)

[themes](#), [8](#)