Package 'editData'

October 7, 2017

Type Package

Title 'RStudio' Addin for Editing a 'data.frame'

Version 0.1.2

Imports shiny (>= 0.13), miniUI (>= 0.1.1), rstudioapi (>= 0.5), DT, tibble

Description An 'RStudio' addin for editing a 'data.frame' or a 'tibble'. You can delete, add or update a 'data.frame'
without coding. You can get resultant data as a 'data.frame'. In the package, modularized 'shiny' app codes are provided.
These modules are intended for reuse across applications.

URL https://github.com/cardiomoon/editData

BugReports https://github.com/cardiomoon/editData/issues

License GPL-3

Encoding UTF-8

Depends R (>= 2.10)

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Keon-Woong Moon [aut, cre]

Maintainer Keon-Woong Moon <cardiomoon@gmail.com>

Repository CRAN

Date/Publication 2017-10-07 15:47:30 UTC

R topics documented:

checkboxInput3 2
dateInput3 3

editableDT 3
editableDTUI 4
editData 5
label3 5
numericInput3 6
radioButtons3 7
sampleData 8
selectInput3 8
textInput3 9

Index **10**

`checkboxInput3` *Create a side-by-side checkboxInput*

Description

Create a side-by-side checkboxInput

Usage

```
checkboxInput3(inputId, label, value = FALSE, width = 100)
```

Arguments

`inputId`	The input slot that will be used to access the value.
`label`	Display label for the control, or NULL for no label.
`value`	Initial value.
`width`	The width of the input in pixel

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         label3("Welcome"),
         checkboxInput3("somevalue", "Some value", FALSE),
         verbatimTextOutput("value")
  )
  server <- function(input, output) {
        output$value <- renderText({ input$somevalue })
  }
  shinyApp(ui, server)
}
```

dateInput3 *Create a side-by-side dateInput*

Description

Create a side-by-side dateInput

Usage

```
dateInput3(inputId, label, width = 100, ...)
```

Arguments

inputId	The input slot that will be used to access the value.
label	Display label for the control, or NULL for no label.
width	The width of the input in pixel
...	arguments to be passed to dateInput

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         label3("Welcome"),
         dateInput3("date", "date"),
         verbatimTextOutput("value")
  )
  server <- function(input, output) {
        output$value <- renderText({ input$date })
  }
  shinyApp(ui, server)
}
```

editableDT *Server function of editData Shiny module*

Description

Server function of editData Shiny module

Usage

```
editableDT(input, output, session, dataname = reactive(""),
  data = reactive(NULL), inputwidth = reactive(100))
```

Arguments

input	input
output	output
session	session
dataname	A string of representing data name
data	A data object
inputwidth	Numeric indicating default input width in pixel

---

editableDTUI *UI of editData Shiny module*

Description

UI of editData Shiny module

Usage

```
editableDTUI(id)
```

Arguments

id	A string

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
ui <- fluidPage(
    textInput("mydata","Enter data name",value="mtcars"),
    editableDTUI("table1"),
    verbatimTextOutput("test"),
    editableDTUI("table2"),
    verbatimTextOutput("test2")
)
server <- function(input, output) {
   df=callModule(editableDT,"table1",dataname=reactive(input$mydata),inputwidth=reactive(170))

    output$test=renderPrint({
         str(df())
    })
    mydf<-editData::sampleData
    df2=callModule(editableDT,"table2",data=reactive(mydf))
    output$test2=renderPrint({
         str(df2())
    })
}
shinyApp(ui, server)
}
```

editData *A shiny app for editing a 'data.frame'*

Description

A shiny app for editing a 'data.frame'

Usage

```
editData(data = NULL, viewer = "dialog")
```

Arguments

data	A tibble or a tbl_df or a data.frame to manipulate
viewer	Specify where the gadget should be displayed. Possible choices are c("dialog","browser","pane")

Value

A manipulated 'data.frame' or NULL

Examples

```
library(shiny)
library(editData)
# Only run examples in interactive R sessions
if (interactive()) {
    result<-editData(mtcars)
    result
}
```

label3 *Create a side-by-side label*

Description

Create a side-by-side label

Usage

```
label3(label, width = 100, bg = NULL, ...)
```

Arguments

label	A text to display
width	The width of the input in pixel
bg	The color of text
...	arguments to be passed to label

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         label3("Welcome"),
         checkboxInput3("somevalue", "Some value", FALSE),
         verbatimTextOutput("value")
  )
  server <- function(input, output) {
        output$value <- renderText({ input$somevalue })
  }
  shinyApp(ui, server)
}
```

`numericInput3` *Create a side-by-side numericInput*

Description

Create a side-by-side numericInput

Usage

```
numericInput3(inputId, label, value, min = NA, max = NA, step = NA,
  width = 100, ...)
```

Arguments

`inputId`	The input slot that will be used to access the value.
`label`	Display label for the control, or NULL for no label.
`value`	Initial value.
`min`	Minimum allowed value
`max`	Maximum allowed value
`step`	Interval to use when stepping between min and max
`width`	The width of the input in pixel
`...`	arguments to be passed to numericInput

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         textInput3("id", "id", ""),
         numericInput3("score","score",value=1)
```

```
    )
    server <- function(input, output) {

    }
    shinyApp(ui, server)
}
```

---

radioButtons3 *Create a side-by-side radioButtons*

Description

Create a side-by-side radioButtons

Usage

```
radioButtons3(inputId, label, choices, bg = NULL, labelwidth = 100,
  inline = FALSE, align = "right", ...)
```

Arguments

inputId	The input slot that will be used to access the value.
label	Display label for the control, or NULL for no label.
choices	List of values to select from
bg	The color of text
labelwidth	The width of the label in pixel
inline	If TRUE, render the choices inline (i.e. horizontally)
align	text align of label
...	arguments to be passed to radioButtons

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         label3("Welcome"),
         radioButtons3("mydata", "mydata", choices=c("mtcars","iris")),
         verbatimTextOutput("value")
  )
  server <- function(input, output) {
        output$value <- renderText({ input$mydata })
  }
  shinyApp(ui, server)
}
```

sampleData *Sample Data for testing 'editData' addin*

Description

A sample dataset containing data for 4 people

Usage

```
sampleData
```

Format

A data.frame with 4 rows and 6 variables:

name Last name

age age in years

country Country Name

sex sex, A factor with two levels.

bloodType Blood Type. A factor with four levels

date Date

selectInput3 *Create a side-by-side selectInput*

Description

Create a side-by-side selectInput

Usage

```
selectInput3(..., width = 100)
```

Arguments

`...`	arguments to be passed to selectInput
`width`	The width of the input in pixel

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         selectInput3("sex", "sex", choices=c("Male","Female")),
        selectInput3("smoking", "smokingStatus", choices=c("Never","Ex-smoker","Smoker"))
    )
    server <- function(input, output) {

    }
    shinyApp(ui, server)
}
```

`textInput3` *Create a side-by-side textInput control for entry of unstructured text values*

Description

Create a side-by-side textInput control for entry of unstructured text values

Usage

```
textInput3(inputId, label, value = "", width = 100, bg = NULL, ...)
```

Arguments

`inputId`	The input slot that will be used to access the value.
`label`	Display label for the control, or NULL for no label.
`value`	Initial value.
`width`	The width of the input in pixel
`bg`	The color of text
`...`	arguments to be passed to textInput

Examples

```
library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
         textInput3("id", "id", ""),
         textInput3("name","name","")
    )
    server <- function(input, output) {

    }
    shinyApp(ui, server)
}
```

Index

∗Topic **datasets**
sampleData, 8

checkboxInput3, 2

dateInput3, 3

editableDT, 3
editableDTUI, 4
editData, 5

label3, 5

numericInput3, 6

radioButtons3, 7

sampleData, 8
selectInput3, 8

textInput3, 9