

Package ‘editData’

October 13, 2022

Type Package

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

Version 0.1.8

Imports shiny (≥ 0.13), miniUI ($\geq 0.1.1$), rstudioapi (≥ 0.5), DT(≥ 0.17), tibble, dplyr, rio, magrittr, shinyWidgets, openxlsx

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 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Keon-Woong Moon [aut, cre]

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

Repository CRAN

Date/Publication 2021-04-02 17:00:02 UTC

R topics documented:

checkboxInput3	2
dateInput3	3

editableDT	4
editableDTUI	4
editData	5
editFiles	6
file2ext	6
label3	6
myget	7
myimport	7
numericInput3	8
pickerInput3	9
radioButtons3	9
sampleData	10
selectInput3	11
selectizeInput3	11
textInput3	12

Index	13
--------------	-----------

checkboxInput3	<i>Create a side-by-side checkboxInput</i>
----------------	--------------------------------------------

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	<i>Server function of editableDT Shiny module</i>
------------	---------------------------------------------------

Description

Server function of editableDT Shiny module

Usage

```
editableDT(input, output, session, data)
```

Arguments

input	input
output	output
session	session
data	A reactive data object

editableDTUI	<i>UI of editableDT Shiny module</i>
--------------	--------------------------------------

Description

UI of editableDT Shiny module

Usage

```
editableDTUI(id)
```

Arguments

id	A string
----	----------

Examples

```
# Only run examples in interactive R sessions
if (interactive()) {
  library(shiny)
  ui=fluidPage(
    selectInput("select", "select", choices=c("mtcars", "iris", "sampleData")),
    textInput("mydata", "mydata", value="mtcars"),
    hr(),
    editableDTUI("editableDT"),
    hr(),
    verbatimTextOutput("test")
  )
}
```

```

server=function(input,output,session){
  data=reactive({
    myget(input$mydata)
  })
  observeEvent(input$select,{
    updateTextInput(session,"mydata",value=input$select)
  })
  result=callModule(editableDT,"editableDT",data=data)
  output$test=renderPrint({
    str(result())
  })
}
shinyApp(ui=ui,server=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", mode = 2)
```

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")
mode	An integer

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
}

```

editFiles	<i>Edit multiple files side by side</i>
-----------	-----------------------------------------

Description

Edit multiple files side by side

Usage

```
editFiles()
```

file2ext	<i>Extract extension from a file name</i>
----------	-------------------------------------------

Description

Extract extension from a file name

Usage

```
file2ext(filename)
```

Arguments

filename	A character string naming a file
----------	----------------------------------

label3	<i>Create a side-by-side label</i>
--------	------------------------------------

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(
    label13("Welcome"),
    checkboxInput3("somevalue", "Some value", FALSE),
    verbatimTextOutput("value")
  )
  server <- function(input, output) {
    output$value <- renderText({ input$somevalue })
  }
  shinyApp(ui, server)
}
```

myget*Return the Value of a Named data.frame*

Description

Return the Value of a Named data.frame

Usage

```
myget(x)
```

Arguments

x Name of data.frame

Examples

```
myget("iris")
myget("mtcars")
```

myimport*Read in a data.frame from a file*

Description

Read in a data.frame from a file

Usage

```
myimport(file, ...)
```

Arguments

file	A character string naming a file
...	Further arguments to be passed to rio::import

numericInput3	<i>Create a side-by-side numericInput</i>
---------------	-------------------------------------------

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)  
  }
```

pickerInput3

Side by side pickerInput

Description

Side by side pickerInput

Usage

```
pickerInput3(...)
```

Arguments

... Further arguments to be passed to pickerInput

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(
    label13("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	<i>Create a side-by-side selectInput</i>
--------------	------------------------------------------

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)
}
```

selectizeInput3	<i>side-by-side selectizeInput</i>
-----------------	------------------------------------

Description

side-by-side selectizeInput

Usage

```
selectizeInput3(..., width = 100)
```

Arguments

...	Further arguments to be passed to selectizeInput
width	Input width in pixel

Examples

```

library(shiny)
# Only run examples in interactive R sessions
if (interactive()) {
  ui <- fluidPage(
    selectizeInput3("color", "color", choices=colors())
  )
  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

- * **datasets**
 - sampleData, 10
- checkboxInput3, 2
- dateInput3, 3
- editableDT, 4
- editableDTUI, 4
- editData, 5
- editFiles, 6
- file2ext, 6
- label3, 6
- myget, 7
- myimport, 7
- numericInput3, 8
- pickerInput3, 9
- radioButtons3, 9
- sampleData, 10
- selectInput3, 11
- selectizeInput3, 11
- textInput3, 12