

Package ‘shinyDatettimePickers’

December 8, 2022

Title Some Datetime Pickers for 'Shiny'

Version 1.0.0

Description Provides three types of datetime pickers for usage in a 'Shiny' UI. A date-time picker is an input field for selecting both a date and a time.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.2

Imports htmltools, shiny, reactR, lubridate

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
Wojciech Maj [cph] ('react-datetime-picker' library
(<https://github.com/wojtekma/react-datetime-picker>)),
Porinn [cph] ('react-datetime-slider-picker' library
(<https://github.com/Porinn/react-datetime-slider-picker>)),
Dmitriy Kovalenko [cph] ('material-ui-pickers' library
(<https://github.com/mui-org/material-ui-pickers>))

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2022-12-08 14:00:02 UTC

R topics documented:

datetimeMaterialPickerInput	2
datetimePickerInput	3
datetimeSliderPickerInput	4

Index	6
--------------	----------

datetimeMaterialPickerInput

Material design datetime picker

Description

A datetime picker for a Shiny UI.

Usage

```
datetimeMaterialPickerInput(  
  inputId,  
  label = NULL,  
  value = NULL,  
  disablePast = FALSE,  
  disableFuture = FALSE,  
  style = NULL  
)
```

Arguments

inputId	the input slot that will be used to access the value
label	a label, a character string (HTML is not allowed), or NULL for no label
value	initial value, either a POSIXct object, or an object coercible to a POSIXct object; if NULL, it is set to the current time
disablePast	logical, whether to disable past dates
disableFuture	logical, whether to disable future dates
style	inline CSS for the container

Value

A shiny.tag object that can be included in a Shiny UI.

Examples

```
if(interactive()){  
  
  library(shinyDatetimePickers)  
  library(shiny)  
  
  ui <- fluidPage(  
    br(),  
    sidebarLayout(  
      sidebarPanel(  
        datetimeMaterialPickerInput(  
          "dtmpicker",  
          label = "Appointment",
```

```
        disablePast = TRUE
      )
    ),
    mainPanel(
      verbatimTextOutput("dtmpicker")
    )
  )
)

server <- function(input, output){
  output[["dtmpicker"]] <- renderPrint({
    input[["dtmpicker"]]
  })
}

shinyApp(ui, server)

}
```

datetimePickerInput *Datetime picker*

Description

A datetime picker for a Shiny UI.

Usage

```
datetimePickerInput(inputId, value = NULL, style = NULL)
```

Arguments

inputId	the input slot that will be used to access the value
value	initial value, either a POSIXct object, or an object coercible to a POSIXct object; if NULL, it is set to the current time
style	inline CSS for the container

Value

A shiny.tag object that can be included in a Shiny UI.

Examples

```
if(interactive()){

  library(shinyDatetimePickers)
  library(shiny)

  ui <- fluidPage(
```

```

br(),
sidebarLayout(
  sidebarPanel(
    tags$fieldset(
      tags$legend("Click to change time"),
      datetimePickerInput(
        "dtpicker",
        style =
          "font-family: Montserrat, 'Segoe UI', Tahoma, sans-serif;"
      )
    )
  ),
  mainPanel(
    verbatimTextOutput("dtpicker")
  )
)
)

server <- function(input, output){
  output[["dtpicker"]] <- renderPrint({
    input[["dtpicker"]]
  })
}

shinyApp(ui, server)
}

```

datetimeSliderPickerInput

Datetime picker with sliders

Description

A datetime picker for a Shiny UI.

Usage

```
datetimeSliderPickerInput(inputId, value = NULL, second = FALSE, save = FALSE)
```

Arguments

inputId	the input slot that will be used to access the value
value	initial value, either a POSIXct object, or an object coercible to a POSIXct object; if NULL, it is set to the current time
second	logical, whether to enable the second picker
save	logical, whether to enable the 'save' button

Value

A shiny.tag object that can be included in a Shiny UI.

Examples

```
if(interactive()){
  library(shinyDatetimePickers)
  library(shiny)

  ui <- fluidPage(
    sidebarLayout(
      sidebarPanel(
        actionButton("setdt", label = as.character(Sys.time()),
                     class = "btn-info")
      ),
      mainPanel()
    )
  )

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

    datetime <- reactiveVal(Sys.time())

    observeEvent(input[["setdt"]], {
      showModal(modalDialog(
        datetimeSliderPickerInput("dtspicker", save = TRUE, value = datetime())
      ))
    })

    observeEvent(input[["dtspicker_save"]], {
      datetime(input[["dtspicker"]])
      removeModal()
      updateActionButton(session, "setdt",
                        label = as.character(input[["dtspicker"]]))
    })

  }

  shinyApp(ui, server)
}
```

Index

`datetimeMaterialPickerInput`, [2](#)
`datetimePickerInput`, [3](#)
`datetimeSliderPickerInput`, [4](#)