

Package: shinyDatettimePickers (via r-universe)

September 3, 2024

Title Some Datetime Pickers for 'Shiny'

Version 1.2.0

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

License GPL-3

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

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

Imports htmltools, lubridate, reactR, shiny

Encoding UTF-8

RoxygenNote 7.2.3

Repository <https://stla.r-universe.dev>

RemoteUrl <https://github.com/stla/shinydatetimepickers>

RemoteRef HEAD

RemoteSha 6cc6b73f71912f616f3daf4098beb76d00c53514

Contents

datetimeMaterialPickerInput	2
datetimePickerInput	3
datetimeSliderPickerInput	4
updateDatetimeMaterialPickerInput	5
updateDatetimePickerInput	6
updateDatetimeSliderPickerInput	7

Index	8
--------------	----------

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)
}
```

updateDatetimeMaterialPickerInput

Update a datetime material picker widget

Description

Change the value of a datetime material picker input.

Usage

```
updateDatetimeMaterialPickerInput(session, inputId, value)
```

Arguments

session	the Shiny session object
inputId	the id of the datetime material picker widget to be updated
value	new value for the datetime material picker widget

Value

No returned value, this just updates the widget.

```
updateDatetimePickerInput
```

Update a datetime picker widget

Description

Change the value of a datetime picker input.

Usage

```
updateDatetimePickerInput(session, inputId, value)
```

Arguments

session	the Shiny session object
inputId	the id of the datetime picker widget to be updated
value	new value for the datetime picker widget

Value

No returned value, this just updates the widget.

updateDatetimeSliderPickerInput

Update a datetime slider picker widget

Description

Change the value of a datetime slider picker input.

Usage

```
updateDatetimeSliderPickerInput(session, inputId, value)
```

Arguments

session	the Shiny session object
inputId	the id of the datetime slider picker widget to be updated
value	new value for the datetime slider picker widget

Value

No returned value, this just updates the widget.

Index

`datetimeMaterialPickerInput`, [2](#)

`datetimePickerInput`, [3](#)

`datetimeSliderPickerInput`, [4](#)

`updateDatetimeMaterialPickerInput`, [5](#)

`updateDatetimePickerInput`, [6](#)

`updateDatetimeSliderPickerInput`, [7](#)