

Package ‘shinyTime’

August 19, 2022

Type Package

Title A Time Input Widget for Shiny

Version 1.0.3

Description Provides a time input widget for Shiny. This widget allows intuitive time input in the '[hh]:[mm]:[ss]' or '[hh]:[mm]' (24H) format by using a separate numeric input for each time component. The interface with R uses date-time objects. See the project page for more information and examples.

License GPL-3 | file LICENSE

Imports htmltools, shiny

URL <https://burgerga.github.io/shinyTime/>,
<https://github.com/burgerga/shinyTime>

BugReports <https://github.com/burgerga/shinyTime/issues>

RoxygenNote 7.2.1

Encoding UTF-8

Language en-US

Suggests testthat (>= 2.1.0), spelling, hms

NeedsCompilation no

Author Gerhard Burger [aut, cre] (<<https://orcid.org/0000-0003-1062-5576>>)

Maintainer Gerhard Burger <burger.ga@gmail.com>

Repository CRAN

Date/Publication 2022-08-19 21:30:02 UTC

R topics documented:

shinyTimeExample	2
timeInput	2
updateTimeInput	3

Index	5
--------------	----------

shinyTimeExample	<i>Show the shinyTime example app</i>
------------------	---------------------------------------

Description

Run a simple shiny app demonstrating the shinyTime functionality.

Usage

```
shinyTimeExample()
```

See Also

Other shinyTime functions: [timeInput\(\)](#), [updateTimeInput\(\)](#)

timeInput	<i>Create a time input</i>
-----------	----------------------------

Description

Creates a time widget that consists of separate numeric inputs for the hours, minutes, and seconds. The input and output values of the time widget are instances of [DateTimeClasses](#), these can be converted to and from character strings with [strptime](#) and [strftime](#). Additionally, the input can be specified as a character string in the 'hh:mm:ss' format or an [hms](#) class. For a simple example app see [shinyTimeExample](#).

Usage

```
timeInput(inputId, label, value = NULL, seconds = TRUE, minute.steps = 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	The desired time value. Must be a instance of DateTimeClasses .
seconds	Show input for seconds. Defaults to TRUE.
minute.steps	Round time to multiples of minute.steps (should be a whole number). If not NULL sets seconds to FALSE.

Value

Returns a POSIXlt object, which can be converted to a POSIXct object with `as.POSIXct` for more efficient storage.

See Also

[strptime](#), [strftime](#), [DateTimeClasses](#)

Other shinyTime functions: [shinyTimeExample\(\)](#), [updateTimeInput\(\)](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- fluidPage(
    # Default value is 00:00:00
    timeInput("time1", "Time:"),

    # Set to current time
    timeInput("time2", "Time:", value = Sys.time()),

    # Set to custom time
    timeInput("time3", "Time:", value = strptime("12:34:56", "%T")),

    # Set to custom time using hms
    timeInput("time4", "Time:", value = hms::as_hms("23:45:07")),

    # Set to custom time using character string
    timeInput("time5", "Time:", value = "21:32:43"),

    # Use hh:mm format
    timeInput("time6", "Time:", seconds = FALSE),

    # Use multiples of 5 minutes
    timeInput("time7", "Time:", minute.steps = 5)
  )

  shinyApp(ui, server = function(input, output) { })
}
```

updateTimeInput

Change a time input on the client

Description

Change the label and/or value of a time input

Usage

```
updateTimeInput(session, inputId, label = NULL, value = NULL)
```

Arguments

session	The session object passed to function given to shinyServer. Default is getDefaultReactiveDomain()
inputId	The id of the input object.
label	The label to set for the input object.
value	The desired time value. Must be an instance of DateTimeClasses .

See Also

Other shinyTime functions: [shinyTimeExample\(\)](#), [timeInput\(\)](#)

Examples

```
## Only run examples in interactive R sessions
if (interactive()) {

  ui <- fluidPage(
    timeInput("time", "Time:"),
    actionButton("to_current_time", "Current time")
  )

  server <- function(input, output, session) {
    observeEvent(input$to_current_time, {
      updateTimeInput(session, "time", value = Sys.time())
    })
  }

  shinyApp(ui, server)
}
```

Index

* shinyTime functions

shinyTimeExample, 2

timeInput, 2

updateTimeInput, 3

DateTimeClasses, 2–4

hms, 2

shinyTimeExample, 2, 2, 3, 4

strftime, 2, 3

strptime, 2, 3

timeInput, 2, 2, 4

updateTimeInput, 2, 3, 3