

Package: shinyDTC (via r-universe)

May 9, 2026

Type Package

Title Simple Dynamic Timer Control

Version 0.1.0

Description A dynamic timer control is a Shiny widget that enables time-based processes in applications. It allows users to execute these processes manually in individual steps or at customizable speeds. The timer can be paused, resumed, or restarted. This control is particularly well-suited for simulations, animations, countdowns, or interactive visualizations.

URL <https://github.com/sigbertklinke/shinyDTC> (development version)

License GPL-3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports rstudioapi, shiny, shinyjs

Suggests knitr, rmarkdown, shinydashboard

VignetteBuilder knitr

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

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

Date/Publication 2025-01-04 15:08:21 UTC

RemoteUrl <https://github.com/sigbertklinke/shinydte>

RemoteRef HEAD

RemoteSha 7648915a035e42061e5e4c67998e8aa003f35bfe

Contents

openApp	2
resetTimer	3
runAppx	3
stepTimer	4
timer	4
timerInput	5
updateTimerInput	6
Index	7

openApp	<i>openApp</i>
---------	----------------

Description

Opens an example Shiny app into the RStudio editor.

Usage

```
openApp(dir = "mini", ...)
```

Arguments

dir	character: name of the app
...	further parameter given to rstudioapi::navigateToFile()

Value

invisibly the result of `navigateToFile`

Examples

```
if (interactive()) run() # runs the minimal example
```

resetTimer	<i>resetTimer</i>
------------	-------------------

Description

Simulates a button press for Reset.

Usage

```
resetTimer(inputId)
```

Arguments

inputId character: input slot

Value

nothing

Examples

```
if (interactive()) vignette("shinyDTC")
```

runAppx	<i>runAppx</i>
---------	----------------

Description

Runs an example Shiny app via [shiny::runExample\(\)](#).

Usage

```
runAppx(example = "mini", x = NULL, ...)
```

Arguments

example character: name of the example to run, or NA to list the available examples
x data frame: data to pass to the app
... further parameter given to [shiny::runExample\(\)](#)

Details

Note that for running a Shiny app more libraries might be necessary.

Value

invisibly the (modified) data set x

Examples

```
if (interactive()) runApp() # runs the minimal example
```

stepTimer	<i>stepTimer</i>
-----------	------------------

Description

Simulates a button press on Step.

Usage

```
stepTimer(inputId)
```

Arguments

inputId	character: input slot
---------	-----------------------

Value

nothing

Examples

```
if (interactive()) vignette("shinyDTC")
```

timer	<i>timer</i>
-------	--------------

Description

A reactive function.

Usage

```
timer(inputId, input, session, ...)
```

Arguments

inputId	character: input slot for a timer
input	an input object
session	a session object
...	further parameters for <code>shiny::reactive()</code>

Value

a reactive function which returns a counter

Examples

```
if (interactive()) vignette("shinyDTC")
```

`timerInput`

timerInput

Description

Constructs a timer widget with a slider and two buttons.

Usage

```
timerInput(inputId, step1 = list(), reset = list(), ...)
```

Arguments

<code>inputId</code>	character: input slot
<code>step1</code>	list: list of parameters for the left (step) button
<code>reset</code>	list: list of parameters for the right (reset) button
<code>...</code>	parameters to the slider

Value

a widget to timer

Examples

```
if (interactive()) vignette("shinyDTC")
```

updateTimerInput	<i>updateTimerInput</i>
------------------	-------------------------

Description

updateTimerInput

Usage

```
updateTimerInput(  
  session = getDefaultReactiveDomain(),  
  inputId,  
  step1 = list(),  
  reset = list(),  
  ...  
)
```

Arguments

session	session object
inputId	id of the input object
step1	modified values for Step button, see shiny::updateActionButton()
reset	modified values for Reset button, see shiny::updateActionButton()
...	modified values for Speed slider, see shiny::updateSliderInput()

Value

nothing

Examples

```
if (interactive()) vignette("shinyDTC")
```

Index

`openApp`, 2

`resetTimer`, 3

`rstudioapi::navigateToFile()`, 2

`runAppx`, 3

`shiny::reactive()`, 4

`shiny::runExample()`, 3

`shiny::updateActionButton()`, 6

`shiny::updateSliderInput()`, 6

`stepTimer`, 4

`timer`, 4

`timerInput`, 5

`updateTimerInput`, 6