Package 'timeR'

October 14, 2022

Type Package

Title Time Your Codes

Version 1.2.0

Author Yifu Yan

Maintainer Yifu Yan <yanyifu94@hotmail.com>

Description Provides a 'timeR' class that makes timing codes easier. One can create 'timeR' objects and use them to record all timings, and extract recordings as data frame for later use.

URL https://github.com/yusuzech/timeR

BugReports https://github.com/yusuzech/timeR/issues

Depends R (>= 3.1.0)

Imports R6, lubridate

License Apache License (== 2.0) | file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-22 18:40:02 UTC

R topics documented:

createTimer *Create a timer object*

Description

Create a timer object

Usage

```
createTimer(verbose = T, precision = "s")
```

Arguments

verbose	A parameter to control whether to print messages while using methods. Default to TRUE.
precision	Precision for time, default to s, valid values are: s,ms and us

Value

a timer object.

Examples

```
timer1 <- createTimer() # print is enabled
timer1 <- createTimer(FALSE) # print is disabled
timer1$start("event1") # start timing for event 1
timer1$stop("event1", comment = "event 1 stopped") # stop timing for event 1(comment is optional)
getTimer(timer1) # get all records in a data frame
```

getTimer *Get the data frame in timer object*

Description

timer object has a built-in data frame that contains all timings. run this function to extract the data frame.

Usage

```
getTimer(object)
```

Arguments

object	The name for timer object.

Value

A data frame containing all records of a timer object.

Examples

```
timer1 <- createTimer()
timer1$start("event1")
Sys.sleep(1)
timer1$stop("event1")
getTimer(timer1)
```

`timeR` *timeR: A package to make timing codes easier*

Description

The timeR package saves your time by timing your code and save recordings to a data frame automatically.So you don't have to do all these steps manually by yourself.

timer is a R6 Class that represent a timer.

Usage

```
timeR
```

Format

An object of class `R6ClassGenerator` of length 24.

Fields

`time` A POSIXct/POSIXlt value of your latest timing.

`event` A string of your latest timing.

`eventTable` A data frame that stores all timings.

`verbose` A printing setting that controls whether to print messages.

Public Methods

`initialize(time,event,verbose,eventTable)` Initialize a timer object. You can also use `createTimer()` function to initialize a timer object.

`start(eventName)` Start timing for a event, `eventName` should be a string

`stop(eventName)` Stop timing for a event.

`getTimer()` Get a data frame that stores all recordings.You can also use `getTimer()` function to get the data frame.

`removeEvent(eventName)` Remove an given row in the eventTable.

`toggleVerbose()` Toggle between `TRUE` and `FALSE` for `verbose`

`getStartTime()` Get start time for a selected event.

`getStopTime()` Get stop time for a selected event.

`getTimeElapsed()` Get time elapsed for a selected event.

`getComment()` Get comment for a selected event.

`getEventf()` Get entire row for a selected event.

`print()` Custom print method for timer class. However, you don't need to use this function to generate custom printing. Custom printing is triggered by default.

Private Methods

`slprint(msg,flag = self$verbose)` A function that controls whether to print extra message.

Examples

```
timer <- createTimer(precision = "ms")
timer$start("event1")
# put some codes in between
timer$stop("event1")

timer$start("event2")
# put some codes in between
timer$stop("event2",comment = "event 2 completed")

table1 <- getTimer(timer)
timer$toggleVerbose() # set verbose to FALSE as default is TRUE

table1 # print all records in a tibble(data frame)

# get attributes for selected events
timer$getStartTime("event1")
timer$getStopTime("event1")
timer$getTimeElapsed("event1")
timer$getComment("event1")
timer$getEvent("event1")
```

Index