

# Package ‘shinytoastr’

October 14, 2022

**Title** Notifications from 'Shiny'

**Version** 2.1.1

**Author** John Papa, Tim Ferrell, Hans Fjällemark, Gábor Csárdi

**Maintainer** Gábor Csárdi <gcsardi@mango-solutions.com>

**Description** Browser notifications in 'Shiny' apps, using  
'toastr': <<https://github.com/CodeSeven/toastr#readme>>.

**License** MIT + file LICENSE

**LazyData** true

**URL** <https://github.com/mangothecat/shinytoastr>

**BugReports** <https://github.com/mangothecat/shinytoastr/issues>

**Imports** shiny

**RoxygenNote** 5.0.1

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-06-06 06:39:52

## R topics documented:

toastr_success . . . . .	2
useToastr . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

`toastr_success`*Create toastr notifications*

---

## Description

There are four functions to create notifications: `toastr_success`, `toastr_info`, `toastr_warning` and `toastr_error`. They have exactly the same arguments and API in general, but they create different kinds of notifications, styled appropriately.

## Usage

```
toastr_success(message, title = "", closeButton = FALSE,
  newestOnTop = FALSE, progressBar = FALSE, position = c("top-right",
  "top-center", "top-left", "top-full-width", "bottom-right", "bottom-center",
  "bottom-left", "bottom-full-width"), preventDuplicates = FALSE,
  showDuration = 300, hideDuration = 1000, timeOut = 5000,
  extendedTimeOut = 1000, showEasing = c("swing", "linear"),
  hideEasing = c("swing", "linear"), showMethod = c("fadeIn", "slideDown",
  "show"), hideMethod = c("fadeOut", "hide"))
```

```
toastr_info(message, title = "", closeButton = FALSE, newestOnTop = FALSE,
  progressBar = FALSE, position = c("top-right", "top-center", "top-left",
  "top-full-width", "bottom-right", "bottom-center", "bottom-left",
  "bottom-full-width"), preventDuplicates = FALSE, showDuration = 300,
  hideDuration = 1000, timeOut = 5000, extendedTimeOut = 1000,
  showEasing = c("swing", "linear"), hideEasing = c("swing", "linear"),
  showMethod = c("fadeIn", "slideDown", "show"), hideMethod = c("fadeOut",
  "hide"))
```

```
toastr_warning(message, title = "", closeButton = FALSE,
  newestOnTop = FALSE, progressBar = FALSE, position = c("top-right",
  "top-center", "top-left", "top-full-width", "bottom-right", "bottom-center",
  "bottom-left", "bottom-full-width"), preventDuplicates = FALSE,
  showDuration = 300, hideDuration = 1000, timeOut = 5000,
  extendedTimeOut = 1000, showEasing = c("swing", "linear"),
  hideEasing = c("swing", "linear"), showMethod = c("fadeIn", "slideDown",
  "show"), hideMethod = c("fadeOut", "hide"))
```

```
toastr_error(message, title = "", closeButton = FALSE,
  newestOnTop = FALSE, progressBar = FALSE, position = c("top-right",
  "top-center", "top-left", "top-full-width", "bottom-right", "bottom-center",
  "bottom-left", "bottom-full-width"), preventDuplicates = FALSE,
  showDuration = 300, hideDuration = 1000, timeOut = 5000,
  extendedTimeOut = 1000, showEasing = c("swing", "linear"),
  hideEasing = c("swing", "linear"), showMethod = c("fadeIn", "slideDown",
  "show"), hideMethod = c("fadeOut", "hide"))
```

**Arguments**

message	Message to show.
title	Optional title, shown on the top.
closeButton	Whether to show a close button. Even if there is a close button, the notification can still be closed by clicking on it. For sticky notifications, it is good practice to show the close button, to tell the user that the notification can be closed.
newestOnTop	Whether to have the newest notification on the top.
progressBar	Whether to show a progress bar.
position	Where to put the notification. Possible values: top-right, top-center, top-left, top-full-width, and the corresponding bottom-right, etc.
preventDuplicates	Whether to prevent showing exactly the same message as the previous one. Note that only the message matters here, the title is ignored.
showDuration	How long the initial show transition should take, in milliseconds.
hideDuration	How long the final hide transition should take, in milliseconds.
timeOut	How long the notification should be kept on the screen, in milliseconds. Set it to zero to keep it on the screen until it is clicked. Note that if the mouse cursor is over the notification, then it is kept on the screen for extendedTimeOut milliseconds, after the cursor has left.
extendedTimeOut	How long to keep the notification on the screen after the mouse cursor leaves it, in milliseconds.
showEasing	Animation easing to show the notification. Possible values: swing, linear.
hideEasing	Animation easing to hide the notification. Possible values: swing, linear.
showMethod	Animation to show the notification. Possible values: fadeIn, slideDown, show.
hideMethod	Animation to hide the notification. Possible values: fadeOut, hide.

**Details**

By default, the notifications disappear automatically after a timeout, unless the mouse cursor is over them.

**Example**

You typically use these functions in the definition of the Shiny server program, as a response to an event, i.e. in the output functions, reactives, or in observe or observeEvent. Here is an example that shows a note when the data was successfully written to a database, and an error otherwise.

```
observeEvent(
  input$save_button,
  {
    tryCatch(
      {
        writeToDB(data)
```

```
      toastr_success("Saved to database")
    },
    error = function(e) {
      toastr_error(title = "Database error", conditionMessage(e))
    }
  }
)
)
```

### See Also

[useToastr](#)

### Examples

```
## See above
```

---

useToastr

*Initialize the toastr notification engine*

---

### Description

Call this function once, from the top of your Shiny UI definition. Here is an example:

```
ui <- shinyUI(fluidPage(
  useToastr(),
  pageWithSidebar(
    headerPanel("Header"),
    sidebarPanel(
      ...
    ),
    mainPanel(
      ...
    )
  )
))
```

### Usage

```
useToastr()
```

### Value

The HTML tags to put into the <head> of the HTML file.

### See Also

[toastr\\_success](#), [toastr\\_info](#), [toastr\\_warning](#), [toastr\\_error](#)

*useToastr*

5

### **Examples**

## See above

# Index

`toastr_error`, [4](#)  
`toastr_error (toastr_success)`, [2](#)  
`toastr_info`, [4](#)  
`toastr_info (toastr_success)`, [2](#)  
`toastr_success`, [2](#), [4](#)  
`toastr_warning`, [4](#)  
`toastr_warning (toastr_success)`, [2](#)  
  
`useToastr`, [4](#), [4](#)