

# Package ‘shinyEffects’

October 14, 2022

**Type** Package

**Title** Customize Your Web Apps with Fancy Effects

**Version** 0.2.0

**Maintainer** David Granjon <dgranjon@gmail.com>

**Description** Add fancy CSS effects to your 'shinydashboards' or 'shiny' apps.  
100% compatible with 'shinydashboardPlus' and 'bs4Dash'.

**License** GPL (>= 2) | file LICENSE

**Imports** htmltools, shiny

**Suggests** shinydashboard, shinydashboardPlus, bs4Dash, rmarkdown, knitr

**URL** <https://github.com/RinteRface/shinyEffects>

**BugReports** <https://github.com/RinteRface/shinyEffects/issues>

**Encoding** UTF-8

**RoxygenNote** 7.1.1.9000

**NeedsCompilation** no

**Author** David Granjon [aut, cre],  
RinteRface [cph]

**Repository** CRAN

**Date/Publication** 2021-05-14 04:10:03 UTC

## R topics documented:

blurContainer . . . . .	2
setBlur . . . . .	2
setPersp . . . . .	3
setPulse . . . . .	4
setShadow . . . . .	6
setShake . . . . .	7
setZoom . . . . .	8
shinyEffectsGallery . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

blurContainer	<i>Blur container</i>
---------------	-----------------------

---

**Description**

Allow to apply a blur effect on a given element.

**Usage**

```
blurContainer(tag)
```

**Arguments**

tag	Element to apply the blur effect.
-----	-----------------------------------

---

setBlur	<i>Custom blur animation</i>
---------	------------------------------

---

**Description**

Allow to apply a blur effect on a given element.

**Usage**

```
setBlur(intensity = 2)
```

**Arguments**

intensity	Blur intensity.
-----------	-----------------

**Examples**

```
if (interactive()) {  
  
  library(shiny)  
  library(shinydashboard)  
  library(shinydashboardPlus)  
  library(shinyEffects)  
  
  boxTag <- box(  
    title = "A box",  
    status = "warning",  
    solidHeader = FALSE,  
    collapsible = TRUE,  
    p("Box Content")  
  )  
}
```

```
shinyApp(  
  ui = dashboardPage(  
    header = dashboardHeader(),  
    sidebar = dashboardSidebar(),  
    body = dashboardBody(  
      setBlur(),  
      fluidRow(blurContainer(boxTag), boxTag)  
    ),  
    controlbar = dashboardControlbar(),  
    title = "DashboardPage"  
  ),  
  server = function(input, output) { }  
)
```

---

setPersp

*Custom perspective effect*

---

## Description

Allow to apply a perspective effect on a given element.

## Usage

```
setPersp(  
  id = NULL,  
  class = NULL,  
  direction = "Y",  
  angle = 45,  
  depth = 600,  
  side = "left",  
  hover = FALSE  
)
```

## Arguments

id	Use this argument if you want to target an individual element.
class	The element to which the persp should be applied.
direction	Perspective direction: "X" or "Y".
angle	Rotation intensity: numeric, 45 by default.
depth	Perspective depth: numeric, 600 by default.
side	Perspective side: "left" or "right".
hover	Whether to apply the persp effect on hover. FALSE by default.

## Examples

```
if (interactive()) {  
  
  library(shiny)  
  library(shinydashboard)  
  library(shinydashboardPlus)  
  library(shinyEffects)  
  
  boxTag <- box(  
    id = "mybox",  
    title = "A box",  
    status = "warning",  
    solidHeader = FALSE,  
    collapsible = TRUE,  
    p("Box Content")  
  )  
  
  shinyApp(  
    ui = dashboardPage(  
      header = dashboardHeader(),  
      sidebar = dashboardSidebar(),  
      body = dashboardBody(  
  
        setPersp(id = "mybox", angle = 45),  
  
        tags$h2("Add Perspective effects"),  
        br(),  
        boxTag  
      ),  
      controlbar = dashboardControlbar(),  
      title = "DashboardPage"  
    ),  
    server = function(input, output) { }  
  )  
}
```

---

setPulse

*Custom pulse animation: UI side*

---

## Description

Allow to apply a pulse animation on a given element.

## Usage

```
setPulse(id = NULL, class = NULL, duration = 1, iteration = 5)
```

**Arguments**

id	Use this argument if you want to target an individual element.
class	The element to which the pulse should be applied. For example, class is set to box. All elements having the box class will be affected.
duration	Pulse frequency. 1s by default.
iteration	Pulse iteration. 5 by default.

**Examples**

```
if (interactive()) {  
  
  library(shiny)  
  library(shinydashboard)  
  library(shinydashboardPlus)  
  library(shinyEffects)  
  
  boxTag <- boxPlus(  
    title = "A box",  
    status = "warning",  
    solidHeader = FALSE,  
    collapsible = TRUE,  
    p("Box Content")  
  )  
  
  shinyApp(  
    ui = dashboardPage(  
      header = dashboardHeader(),  
      sidebar = dashboardSidebar(),  
      body = dashboardBody(  
  
        setPulse(class = "box", duration = 1, iteration = 100),  
        setPulse(id = "my-progress", duration = 1, iteration = 100),  
  
        tags$h2("Add pulse animation to the box class"),  
        fluidRow(boxTag, boxTag),  
        tags$h2("Add pulse animation only to the first element using id"),  
        tagAppendAttributes(  
          progressBar(  
            value = 10,  
            striped = TRUE,  
            animated = TRUE,  
            vertical = TRUE  
          ),  
          id = "my-progress"  
        ),  
        progressBar(  
          value = 50,  
          status = "warning",  
          size = "xs",  
          vertical = TRUE  
        )  
      )  
    )  
  )  
}
```

```

    ),
    controlbar = dashboardControlbar(),
    title = "DashboardPage"
  ),
  server = function(input, output) { }
)
}

```

---

setShadow

*Custom shadows*


---

## Description

Allow to apply a shadow on a given element.

## Usage

```
setShadow(id = NULL, class = NULL)
```

## Arguments

<code>id</code>	Use this argument if you want to target an individual element.
<code>class</code>	The element to which the shadow should be applied. For example, class is set to box.

## Examples

```

if (interactive()) {

  library(shiny)
  library(shinydashboard)
  library(shinydashboardPlus)
  library(shinyEffects)

  boxTag <- box(
    title = "A box",
    status = "warning",
    solidHeader = FALSE,
    collapsible = TRUE,
    p("Box Content")
  )

  shinyApp(
    ui = dashboardPage(
      header = dashboardHeader(),
      sidebar = dashboardSidebar(),
      body = dashboardBody(

        setShadow(class = "box"),
        setShadow(id = "my-progress"),

```

```
tags$h2("Add shadow to the box class"),
fluidRow(boxTag, boxTag),
tags$h2("Add shadow only to the first element using id"),
tagAppendAttributes(
  progressBar(
    value = 10,
    striped = TRUE,
    animated = TRUE,
    vertical = TRUE
  ),
  id = "my-progress"
),
progressBar(
  value = 50,
  status = "warning",
  size = "xs",
)
),
controlbar = dashboardControlbar(),
title = "DashboardPage"
),
server = function(input, output) { }
}
```

---

setShake

*Custom shake animation*

---

## Description

Allow to apply a shake animation on a given element.

## Usage

```
setShake(id = NULL, class = NULL, duration = 0.82)
```

## Arguments

id	Use this argument if you want to target an individual element.
class	The element to which the shake should be applied. For example, class is set to box.
duration	Shake total duration. 0.82s by default.

## Examples

```
if (interactive()) {
  library(shiny)
```

```

library(shinydashboard)
library(shinydashboardPlus)
library(shinyEffects)

boxTag <- box(
  title = "A box",
  status = "warning",
  solidHeader = FALSE,
  collapsible = TRUE,
  p("Box Content")
)

shinyApp(
  ui = dashboardPage(
    header = dashboardHeader(),
    sidebar = dashboardSidebar(),
    body = dashboardBody(

      setShake(class = "box"),
      setShake(id = "my-progress"),

      tags$h2("Add shake to the box class"),
      fluidRow(boxTag, boxTag),
      tags$h2("Add shake only to the first element using id"),
      tagAppendAttributes(
        progressBar(
          value = 10,
          striped = TRUE,
          animated = TRUE,
          vertical = TRUE
        ),
        id = "my-progress"
      ),
      progressBar(
        value = 50,
        status = "warning",
        size = "xs",
      )
    ),
    controlbar = dashboardControlbar(),
    title = "Shake Effect"
  ),
  server = function(input, output) { }
)
}

```

---

setZoom

*Custom zoom*


---

### Description

Allow to apply a zoom on a given element.



**Usage**

```
setZoom(id = NULL, class = NULL, scale = 1.05)
```

**Arguments**

<code>id</code>	Use this argument if you want to target an individual element.
<code>class</code>	The element to which the zoom should be applied. For example, class is set to box.
<code>scale</code>	Zoom scale. 1.05 by default (5% bigger than the normal). We do not recommend using higher values, except in some particular cases.

**Examples**

```
if (interactive()) {  
  
  library(shiny)  
  library(shinydashboard)  
  library(shinydashboardPlus)  
  library(shinyEffects)  
  
  boxTag <- box(  
    title = "A box",  
    status = "warning",  
    solidHeader = FALSE,  
    collapsible = TRUE,  
    p("Box Content")  
  )  
  
  shinyApp(  
    ui = dashboardPage(  
      header = dashboardHeader(),  
      sidebar = dashboardSidebar(),  
      body = dashboardBody(  
  
        setZoom(class = "box"),  
        setZoom(id = "my-progress"),  
  
        tags$h2("Add zoom to the box class"),  
        fluidRow(boxTag, boxTag),  
        tags$h2("Add zoom only to the first element using id"),  
        tagAppendAttributes(  
          progressBar(  
            value = 10,  
            striped = TRUE,  
            animated = TRUE,  
            vertical = TRUE  
          ),  
          id = "my-progress"  
        ),  
        progressBar(  
          value = 50,  

```

```
        status = "warning",
        size = "xs",
      )
    ),
    controlbar = dashboardControlbar(),
    title = "DashboardPage"
  ),
  server = function(input, output) { }
}
```

---

shinyEffectsGallery    *Launch the shinyEffects Gallery*

---

### **Description**

A gallery of all components available in shinyEffects.

### **Usage**

```
shinyEffectsGallery()
```

### **Examples**

```
if (interactive()) {
  shinyEffectsGallery()
}
```

# Index

[blurContainer](#), 2

[setBlur](#), 2

[setPersp](#), 3

[setPulse](#), 4

[setShadow](#), 6

[setShake](#), 7

[setZoom](#), 8

[shinyEffectsGallery](#), 10