Package ‘waiter’

May 5, 2021

Title Loading Screen for ‘Shiny’
Version 0.2.1
Date 2021-05-02
Description
Full screen and partial loading screens for ‘Shiny’ with spinners, progress bars, and notifications.
License MIT + file LICENSE
URL https://waiter.john-coene.com/,
https://github.com/JohnCoene/waiter
BugReports https://github.com/JohnCoene/waiter/issues
Encoding UTF-8
Imports R6, shiny, crayon, magrittr, htmltools
RoxygenNote 7.1.1
Suggests knitr, rmarkdown
VignetteBuilder knitr
NeedsCompilation no
Author John Coene [aut, cre],
Jinhwan Kim [ctb],
Victor Granda [ctb] (<https://orcid.org/0000-0002-0469-1991>)
Maintainer John Coene <jcoenep@gmail.com>
Repository CRAN
Date/Publication 2021-05-05 18:10:02 UTC

R topics documented:

  garcon ................................................................. 2
  hostess .............................................................. 5
  hostessLoader ......................................................... 9
  preview_spinner ..................................................... 12
  spinners ............................................................ 12
Description

Create a garcon to animate images on the waiter.

Usage

use_garcon()

Methods

Public methods:
  • Garcon$new()
  • Garcon$set()
  • Garcon$inc()
  • Garcon$reset()
  • Garcon$destroy()
  • Garcon$print()
  • Garcon$close()
  • Garcon$clone()

Method new():

Usage:
Garcon$new(
  image,
  bg_color = "#FFFFFF",
  opacity = 0.5,
  direction = c("bt", "tb", "lr", "rl"),
  filter = NULL
)

Arguments:
  image  The CSS id of the image tag.
  bg_color  Background overlay color in hexadecimal or RGB.
  opacity  Overlay transparency.
direction  Animation direction. Possible values: lr (left to right), rl (right to left), bt (bottom to top), tb (top to bottom).
filter  Filter to apply, options are blur, grayscale, sepia, hue-rotate, invert, opacity.
Details: Initialise the garçon.
Examples:
\dontrun{Garcon$new("img")$set(30)}

Method set():
Usage:
Garcon$set(value)
Arguments:
value  Percentage to set to.
Details: Value to set the garçon to.
Examples:
\dontrun{Garcon$new("img")$set(30)}

Method inc():
Usage:
Garcon$inc(value)
Arguments:
value  Percentage to increase to.
Details: Value to increase the garçon to.
Examples:
\dontrun{Garcon$new("img")$inc(30)}

Method reset():
Usage:
Garcon$reset(value)
Arguments:
value  Percentage to set to.
Details: Reset the garçon to.
Examples:
\dontrun{Garcon$new("img")$set(30)$reset()}

Method destroy():
Usage:
Garcon$destroy()
Details: Kill the garçon to.
Examples:
\dontrun{Garcon$new("img")$set(30)$destroy()}
Method `print()`:  
 Usage:  
 Garcon$print()  
 Details: print the garçon

Method `close()`:  
 Usage:  
 Garcon$close()  
 Details: Close the garçon. 
 Examples:  
 \dontrun{Garcon$new("img")$set(30)$close()}

Method `clone()`: The objects of this class are cloneable with this method.  
 Usage:  
 Garcon$clone(deep = FALSE)  
 Arguments:  
 deep Whether to make a deep clone.

Examples

```r
# Method `Garcon$new`
#-----------------------------------------------
# Not run: Garcon$new("img")$set(30)

# Method `Garcon$set`
#-----------------------------------------------
# Not run: Garcon$new("img")$set(30)

# Method `Garcon$inc`
#-----------------------------------------------
# Not run: Garcon$new("img")$inc(30)

# Method `Garcon$reset`
#-----------------------------------------------
# Not run: Garcon$new("img")$set(30)$reset()

# Method `Garcon$destroy`
#-----------------------------------------------
```
Description

Add hostess dependencies.

Usage

use_hostess()

Methods

Public methods:
- Hostess$new()
- Hostess$start()
- Hostess$print()
- Hostess$set()
- Hostess$inc()
- Hostess$close()
- Hostess$get_loader()
- Hostess$set_loader()
- Hostess$notify()
- Hostess$clone()

Method new():

Usage:
Hostess$new(id = NULL, min = 0, max = 100, n = 1, infinite = FALSE)

Arguments:
- id: Id used in hostess_loader if you generate the loader with the loader method you may leave this NULL.
- min, max: Minimum and maximum representing the starting and ending points of the progress bar.
- n: Number of loaders to generate.
- infinite: Set to TRUE to create a never ending loading bar, ideal when you cannot compute increments or assess the time it might take before the loading bar should be removed.
Details: Create a hostess.

Examples:
\dontrun{Hostess\$new()}

**Method** `start()`:

**Usage:**
Hostess\$start()

**Details:** Start the hostess

**Method** `print()`:

**Usage:**
Hostess\$print()

**Details:** Print the hostess

**Method** `set()`:

**Usage:**
Hostess\$set(value)

**Arguments:**
value  Value to set, between 0 and 100.

**Details:** Set the hostess loading bar.

**Examples:**
\dontrun{Hostess\$new()\$set(20)}

**Method** `inc()`:

**Usage:**
Hostess\$inc(value)

**Arguments:**
value  Value to set, between 0 and 100.

**Details:** Increase the hostess loading bar.

**Examples:**
\dontrun{Hostess\$new()\$inc(10)}

**Method** `close()`:

**Usage:**
Hostess\$close()

**Details:** Close the hostess

**Examples:**
\dontrun{Waitress\$new("#plot")\$close()}

**Method** `get_loader()`:

**Usage:**
Hostess$\text{get\_loader}(
    \text{preset} = \text{NULL},
    \text{text\_color} = "\text{#FFFFFF}",
    \text{center\_page} = \text{FALSE},
    \text{class} = "",
    \text{min} = \text{NULL},
    \text{max} = \text{NULL},
    \text{svg} = \text{NULL},
    \text{progress\_type} = c("stroke", "fill"),
    \text{fill\_direction} = c("btt", "ttb", "ltr", "rtl"),
    \text{stroke\_direction} = c("normal", "reverse"),
    \text{fill\_color} = \text{NULL},
    \text{stroke\_color} = \text{NULL},
    ...)

\text{Arguments:}
\begin{itemize}
    \item \text{preset} \ A \ loading \ bar \ preset, \ see \ section \ below.
    \item \text{text\_color} \ The \ color \ of \ the \ loading \ text.
    \item \text{center\_page} \ By \ default \ the \ hostess \ is \ centered \ in \ the \ middle \ of \ the \ screen, \ ideal \ when \ using \ it \ with \ waiter \ full \ screen, \ set \ to \ \text{FALSE} \ to \ prevent \ that.
    \item \text{class} \ CSS \ class.
    \item \text{min, max} \ Minimum \ and \ maximum \ representing \ the \ starting \ and \ ending \ points \ of \ the \ progress \ bar.
    \item \text{svg} \ Either \ an \ svg \ path \ e.g.: \text{M10 10L90 10} \ or \ the \ path \ to \ a \ .svg \ file. \ Note \ that \ if \ passing \ the \ latter \ it \ must \ be \ made \ available \ to \ Shiny \ by \ placing \ it \ either \ in \ the \ www \ folder \ or \ using \ \text{shiny::addResourcePath}().
    \item \text{progress\_type} \ The \ progress \ type, \ either \ stroke \ or \ fill. \ The \ former \ traces \ the \ path \ of \ the \ svg \ while \ the \ latter \ fills \ it \ progressively.
    \item \text{fill\_direction, stroke\_direction} \ The \ direction \ which \ the \ progress \ bar \ should \ take. \ Whether \ fill\_direction \ or \ stroke\_direction \ is \ used \ depends \ on \ \text{progress\_type}.
    \item \text{fill\_color, stroke\_color} \ The \ color \ to \ use \ for \ the \ progress \ bar. \ Whether \ fill\_color \ or \ stroke\_color \ is \ used \ depends \ on \ \text{progress\_type}.
\end{itemize}
\text{...} \ Any \ other \ other \ advanced \ options \ to \ pass \ to \ the \ \text{loaded} \ see \ the \ \text{official documentation}.

\text{Details:} \ Create \ a \ hostess \ loading \ bar.

\text{Examples:}
\dontrun{Hostess$new()$get\_loader()}

\textbf{Method} \ \text{set\_loader}():

\textbf{Usage:}
Hostess$set\_loader(loader)

\textbf{Arguments:}
\begin{itemize}
    \item \text{loader} \ Loader \ as \ defined \ by \ \text{hostess\_loader}().
\end{itemize}

\textbf{Details:} \ Set \ a \ hostess \ loader \ as \ defined \ by \ \text{hostess\_loader}().

\textbf{Examples:}
```r
\dontrun{
  loader <- hostess_loader()
  Hostess$new()$set_loader(loader)
}

Method notify():

Usage:
Hostess$notify(
  html = NULL,
  background_color = "transparent",
  text_color = "black",
  position = c("br", "tr", "bl", "tl")
)

Arguments:
  html Additional HTML content of the tag or a character string.
  background_color Background color of the notification.
  text_color Color of text of html.
  position Position of the notification on the screen. Where br is the bottom-right, tr is the
              top-right, bl is bottom-left, and tl is the top-left.

Details: Use the hostess as a notification. It is hidden when set tpo 100.

Examples:
\dontrun{Hostess$new()$notify()}

Method clone(): The objects of this class are cloneable with this method.

Usage:
Hostess$clone(deep = FALSE)

Arguments:
  deep Whether to make a deep clone.

Examples
```
hostessLoader

## Not run: Hostess$new()$inc(10)

## Method `Hostess$close`

## Not run: Waitress$new("#plot")$close()

## Method `Hostess$get_loader`

## Not run: Hostess$new()$get_loader()

## Method `Hostess$set_loader`

## Not run:
loader <- hostess_loader()
Hostess$new()$set_loader(loader)

## End(Not run)

## Method `Hostess$notify`

## Not run: Hostess$new()$notify()

---

**hostessLoader**  

**Loader**

---

**Description**

Customise the Hostess loading bar.

**Usage**

```r
hostess_loader(
  id = "hostess",
  preset = NULL,
  text_color = "#FFFFFF",
  center_page = FALSE,
  class = "",
  min = 0,
  max = 100,
  svg = NULL,
  progress_type = c("stroke", "fill"),
)```
```r
hostessLoader

fill_direction = c("btt", "ttb", "ltr", "rtl"),
stroke_direction = c("normal", "reverse"),
fill_color = NULL,
stroke_color = NULL,
...
)

hostess_gradient(angle = 0, duration = 1, colors = c("red", "white", "blue"))

hostess_bubble(
  color_background = "#697682",
  color_bubble = "#f7fff7",
  count = 25,
  duration = 1
)

hostess_stripe(color1 = "#697682", color2 = "#f7fff7", duration = 1)

Arguments

id            Id of hostess (valid CSS).
preset        A loading bar preset, see section below.
text_color    The color of the loading text.
center_page   By default the hostess is not centered in the middle of the screen, centering in 
               the middle of the page is however ideal when using it with waiter full screen, for 
               the latter set to TRUE.
class         CSS class.
min, max      Minimum and maximum representing the starting and ending points of the progress 
               bar.
svg           Either an svg path e.g.: M10 10L90 10 or the path to a .svg file. Note that if 
               passing the latter it must be made available to Shiny by placing it either in the 
               www folder or using shiny::addResourcePath().
progress_type The progress type, either stroke or fill. The former traces the path of the 
                svg while the latter fills it progressively.
fill_direction, stroke_direction
               The direction which the progress bar should take. Whether fill_direction or 
               stroke_direction is used depends on progress_type.
fill_color, stroke_color
               The color to use for the progress bar. Whether fill_color or stroke_color is 
               used depends on progress_type.

...            Any other other advanced options to pass to the loaded see the official documentation.
angle         Angle of gradient.
duration      Duration of the loop.
colors         Color vectors composing the gradient.
```
color_background  The background of the color.
color_bubble     The color of the bubbles contour.
count           The number of bubbles.
color1, color2  Colors of stripes.

Presets

• line
• fan
• circle
• bubble
• rainbow
• energy
• stripe
• text

Examples

library(shiny)
library(waiter)

# diagonal line
path <- "M10 10L90 30"

ui <- fluidPage(
  use_waiter(),
  use_hostess(),
  actionButton("draw", "redraw"),
  plotOutput("plot")
)

server <- function(input, output) {

  dataset <- reactive({
    input$draw

    hostess <- Hostess$new(min = 0, max = 10)
    hostess$set_loader <- hostess_loader(
      progress_type = "stroke",
      stroke_color = hostess_stripe()
    )
    waiter <- Waiter$new(
      "plot",
      hostess$loader()
    )

    waiter$show()
  })

  output$plot <- renderPlot({
    lines(path, col = "red")
  })
}

server(input, output)
for(i in 1:10){
  Sys.sleep(.2)
  hostess$inc(i)
}
runif(100)
}

output$plot <- renderPlot(plot(dataset()))

if(interactive()) shinyApp(ui, server)

---

**preview_spinner**  
*Preview spinner*

---

**Description**

Allows previewing spinners in web browser or RStudio Viewer.

**Usage**

`preview_spinner(spinner, bg_color = "black")`

**Arguments**

- **spinner**  
  A waiter `link{spinner}`.
- **bg_color**  
  Background color.

**Examples**

`if(interactive()) preview_spinner(spin_1())`

---

**spinners**  
*Spinners*

---

**Description**

Spinkit spinners to use with `waiter_show`. 
spinners

Usage

spin_rotating_plane()
spin_fading_circles()
spin_folding_cube()
spin_double_bounce()
spin_wave()
spin_wandering_cubes()
spin_pulse()
spin_chasing_dots()
spin_three_bounce()
spin_circle()
spin_rotate()
spin_solar()
spin_orbit()
spin_squares()
spin_cube_grid()
spin_circles()
spin_orbiter()
spin_pixel()
spin_flower()
spin_dual_ring()
spin_heart()
spin_ellipsis()
spin_facebook()
spin_hourglass()
spin_ring()
spin_ripple()
spin_terminal()
spin_loader()
spin_throbber()
spin_refresh()
spin_heartbeat()
spin_gauge()
spin_3k()
spin_wobblebar()
spin_atebits()
spin_whirly()
spin_flowers()
spin_dots()
spin_3circles()
spin_plus()
spin_pulsar()
spin_hexdots()
spin_inner_circles()
spin_pong()
spin_timer()
spin_ball()
spin_dual_circle()
spin_seven_circle()
spinners

spin_clock()
spin_pushing_shapes()
spin_fill()
spin_rhombus()
spin_balance()
spin_square_circle()
spin_circle_square()
spin_puzzle()
spin_half()
spin_loaders(id = 1, color = "white", style = NULL)
spin_1()
spin_2()
spin_3()
spin_4()
spin_5()
spin_6()
spin_google()

Arguments

id       The spinner identifier, an integer between 1, and 42.
color    Desired color of spinner.
style    CSS style to apply to spinner.

Details

Much of the CSS used is to provide those spinners. One can greatly reduce the load on the browser by only sourcing the CSS for the spinners required. You can find out which CSS kits are required to load by using the spinner in the R console as shown in the example. This prints the kit and instructions to only source the required file.
Value

An object of class spinner.

Examples

spin_rotating_plane()

transient

Description

A convenience function to create a waiter with transparent background.

Usage

transparent(alpha = 0)

Arguments

alpha Alpha channel where 0 is completely transparent and 1 is opaque.

Examples

transparent()

use_steward

Description

A colorful steward to work with the waiter.

Usage

use_steward(
    colors = c("#ee7752", "#e73c7e", "#23a6d5", "#23d5ab"),
    speed = 30,
    angle = -45
)

Arguments

colors Color palette forming gradient.
speed Seconds it takes to loop over colors.
angle Degrees at which colors slide.
Description

Programatically show and hide loading screens.

Usage

use_waiter(spinners = 1:7, include_js = TRUE)

waiter_use(spinners = 1:7, include_js = TRUE)

waiter_show(
  id = NULL,
  html = spin_1(),
  color = "#333e48",
  logo = "",
  hide_on_render = !is.null(id)
)

waiter_show_on_load(html = spin_1(), color = "#333e48")

waiter_preloader(html = spin_1(), color = "#333e48")

waiter_hide_on_render(id)

waiter_on_busy(html = spin_1(), color = "#333e48", logo = "")

waiter_hide(id = NULL)

waiter_update(id = NULL, html = NULL)

Arguments

spinners  Spinners to include. By default all the CSS files for all spinners are included you can customise this only that which you need in order to reduce the amount of CSS that needs to be loaded and improve page loading speed. There are 7 spinner kits. The spinner kit required for the spinner you use is printed in the R console when using the spinner. You can specify a single spinner kit e.g.: 1 or multiple spinner kits as a vector e.g.: c(1,3,6).

include_js  Deprecated argument, no longer needed.

id  Id of element to hide or element on which to show waiter over.

html  HTML content of waiter, generally a spinner, see spinners.

color  Background color of loading screen.

logo  Logo to display.
hide_on_render  Set to TRUE to automatically hide the waiter when the plot in id is drawn. Note the latter will only work with shiny plots, tables, htmlwidgets, etc. but will not work with arbitrary elements.

Functions

- use_waiter and waiter_use: waiter dependencies to include anywhere in your UI but ideally at the top.
- waiter_show_on_load: Show a waiter on page load, before the session is even loaded, include in UI after use_waiter.
- waiter_show: Show waiting screen.
- waiter_hide: Hide any waiting screen.
- waiter_on_busy: Automatically shows the waiting screen when the server is busy, and hides it when it goes back to idle.
- waiter_update: Update the content html of the waiting screen.
- waiter_hide_on_render: Hide any waiting screen when the output is drawn, useful for outputs that take a long time to draw, use in ui.
- waiter_preloader: Shows the waiter on load and automatically removes it once all the UI is rendered, only runs on the first load of the app.

Examples

library(shiny)

ui <- fluidPage(
  use_waiter(),  # dependencies
  waiter_show_on_load(spin_fading_circles()),  # shows before anything else
  actionButton("show", "Show loading for 5 seconds")
)

server <- function(input, output, session){
  waiter_hide()  # will hide *on_load waiter

  observeEvent(input$show, {
    waiter_show(
      html = tagList(
        spin_fading_circles(),
        "Loading ..."
      )
    )
    Sys.sleep(3)
    waiter_hide()
  })
}

if(interactive()) shinyApp(ui, server)
Description

Create a waiter to then show, hide or update its content.

Details

Create an object to show a waiting screen on either the entire application or just a portion of the app by specifying the id. Then show, then hide or meanwhile update the content of the waiter.

Methods

Public methods:

• `Waiter$new()`
• `Waiter$show()`
• `Waiter$hide()`
• `Waiter$update()`
• `Waiter$print()`
• `Waiter$clone()`

Method `new()`:

Usage:

```r
Waiter$new(  
id = NULL,
html = NULL,
color = NULL,
logo = NULL,
hide_on_render = !is.null(id),
hide_on_error = !is.null(id),
hide_on_silent_error = !is.null(id)
)
```

Arguments:

id  Id, or vector of ids, of element on which to overlay the waiter, if NULL the waiter is applied to the entire body.
html HTML content of waiter, generally a spinner, see `spinners` or a list of the latter.
color Background color of loading screen.
logo Logo to display.
hide_on_render Set to TRUE to automatically hide the waiter when the element in id is drawn.
 Note the latter will work with shiny plots, tables, htmlwidgets, etc. but will not work with arbitrary elements.
hide_on_error, hide_on_silent_error Whether to hide the waiter when the underlying element throws an error. Silent error are thrown by `req` and `validate`.

Details: Create a waiter.

Examples:
\dontrun{Waiter$new()}

Method show():
Usage:
Waiter$show()
Details: Show the waiter.

Method hide():
Usage:
Waiter$hide()
Details: Hide the waiter.

Method update():
Usage:
Waiter$update(html = NULL)
Arguments:
h HTML content of waiter, generally a spinner, see spinners.
Details: Update the waiter’s html content.

Method print():
Usage:
Waiter$print()
Details: print the waiter

Method clone(): The objects of this class are cloneable with this method.
Usage:
Waiter$clone(deep = FALSE)
Arguments:
dep Whether to make a deep clone.

Examples

## Method 'Waiter$new'
```r
## Not run: Waiter$new()
```
### waiterTheme  

**Define a Theme**

**Description**

Define a theme to be used by all waiter loading screens. These can be overridden in individual loading screens.

**Usage**

```r
waiter_set_theme(html = spin_1(), color = "#333e48", logo = "")

waiter_get_theme()

waiter_unset_theme()
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>html</td>
<td>HTML content of waiter, generally a spinner, see spinners.</td>
</tr>
<tr>
<td>color</td>
<td>Background color of loading screen.</td>
</tr>
<tr>
<td>logo</td>
<td>Logo to display.</td>
</tr>
</tbody>
</table>

### waitress  

**Waitress**

**Description**

Programatically show and hide loading bars.

**Usage**

```r
use_waitress(color = "#697682", percent_color = "#333333")
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument, percent_color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>color, percent_color</td>
<td>Color of waitress and color of percent text shown when theme is set to overlay-percent.</td>
</tr>
</tbody>
</table>

**Details**

You can pipe the methods with `. Waitress$new()` and `call_waitress()` are equivalent.
Examples

```r
library(shiny)

ui <- fluidPage(
  use_waitress("red"), # dependencies
  sliderInput("set", "percentage", 1, 100, step = 5, value = 1)
)

server <- function(input, output, session){

  w <- Waitress$# call a waitress
  start() # start waitress

  observeEvent(input$set, {
    w$set(input$set) # set at percentage
  })
}

if(interactive()) shinyApp(ui, server)
```

waitressClass  

**Waitress R6 Class**

Description

Create a waitress (progress bar) and programmatically set or increase its percentage, then hide it when done.

Methods

**Public methods:**

- `Waitress$\text{new}()`
- `Waitress$\text{start}()`
- `Waitress$\text{notify}()`
- `Waitress$\text{set}()`
- `Waitress$\text{auto}()`
- `Waitress$\text{inc}()`
- `Waitress$\text{close}()`
- `Waitress$\text{print}()`
- `Waitress$\text{clone}()`

**Method** `\text{new}()`:

*Usage:*
Waitress$new(
  selector = NULL,
  theme = c("line", "overlay", "overlay-radius", "overlay-opacity", "overlay-percent"),
  min = 0,
  max = 100,
  infinite = FALSE,
  hide_on_render = FALSE
)

Arguments:

selector Element selector to apply the waitress to, if NULL then the waitress is applied to the whole screen.
theme A valid theme, see function usage.
min, max Minimum and maximum representing the starting and ending points of the progress bar.
infinite Set to TRUE to create a never ending loading bar, ideal when you cannot compute increments or assess the time it might take before the loading bar should be removed.
hide_on_render Set to TRUE to automatically hide the waitress when the element in id is rendered. Note the latter will work with shiny plots, tables, htmlwidgets, etc. but will not work with arbitrary elements.

color, percent_color Color of waitress and color of percent text shown when theme is set to overlay-percent.

Details: Create a waitress.

Examples:
\dontrun{Waitress$new("#plot")}

Method start():

Usage:
Waitress$start(
  html = NULL,
  background_color = "transparent",
  text_color = "black"
)

Arguments:

html HTML content to show over the waitress, accepts htmltools and shiny tags.
background_color The background color of the html.
text_color The color of the html content.

Details: Start the waitress.

Examples:
\dontrun{Waitress$new("#plot")$start()}

Method notify():

Usage:
Waitress$notify(
    html = NULL,
    background_color = "white",
    text_color = "black",
    position = c("br", "tr", "bl", "tl")
)

Arguments:
html HTML content to show over the waitress, accepts htmltools and shiny tags.
background_color The background color of the html.
text_color The color of the html content.
position Position of the notification on the screen. Where br is the bottom-right, tr is the top-right, bl is bottom-left, and tl is the top-left.

Details: Show the waitress as a notification.
Examples:
\dontrun{Waitress$new()$notify()}

Method set():
Usage:
Waitress$set(value)
Arguments:
value Value to set waitress to.

Details: Set the waitress to a specific percentage.
Examples:
\dontrun{Waitress$new("#plot")$set(20)}

Method auto():
Usage:
Waitress$auto(value, ms)
Arguments:
value Value to set waitress to.
ms Number of Milliseconds

Details: Automatically start and end the waitress.
Examples:
\dontrun{Waitress$new("#plot")$auto(20, 2000)}

Method inc():
Usage:
Waitress$inc(value)
Arguments:
value Value to increase waitress to.

Details: Increase the waitress by a percentage.
Examples:
\dontrun{Waitress$new("#plot")$inc(30)}

**Method** `close()`:
*Usage:* 
`Waitress$close()`
*Details:* Close the waitress.
*Examples:*
\dontrun{Waitress$new("#plot")$close()}

**Method** `print()`:
*Usage:* 
`Waitress$print()`
*Details:* Print the waitress.
*Examples:*
\dontrun{Waitress$new("#plot")$hide()}

**Method** `clone()`: The objects of this class are cloneable with this method.
*Usage:* 
`Waitress$clone(deep = FALSE)`
*Arguments:*
  deep Whether to make a deep clone.

Examples

```r
## Method `Waitress$new`
## ------------------------
## Not run: Waitress$new("#plot")
##------------------------
## Method `Waitress$start`
## ------------------------
## Not run: Waitress$new("#plot")$start()
##------------------------
## Method `Waitress$notify`
## ------------------------
## Not run: Waitress$new()$notify()
##------------------------
## Method `Waitress$set`
##------------------------
```
## Not run: Waitress$new("#plot")$set(20)

## Method `Waitress$auto`

## Not run: Waitress$new("#plot")$auto(20, 2000)

## Method `Waitress$inc`

## Not run: Waitress$new("#plot")$inc(30)

## Method `Waitress$close`

## Not run: Waitress$new("#plot")$close()

## Method `Waitress$print`

## Not run: Waitress$new("#plot")$hide()
Garcon (garcon), 2
garcon, 2

Hostess (hostess), 5
hostess, 5
hostess_bubble (hostessLoader), 9
hostess_gradient (hostessLoader), 9
hostess_loader (hostessLoader), 9
hostess_loader(), 7
hostess_stripe (hostessLoader), 9
hostessLoader, 9

preview_spinner, 12

req, 19

shiny::addResourcePath(), 7, 10
spin_1 (spinners), 12
spin_2 (spinners), 12
spin_3 (spinners), 12
spin_3circles (spinners), 12
spin_3k (spinners), 12
spin_4 (spinners), 12
spin_5 (spinners), 12
spin_6 (spinners), 12
spin_atebits (spinners), 12
spin_balance (spinners), 12
spin_ball (spinners), 12
spin_chasing_dots (spinners), 12
spin_circle (spinners), 12
spin_circle_square (spinners), 12
spin_circles (spinners), 12
spin_clock (spinners), 12
spin_cube_grid (spinners), 12
spin_dots (spinners), 12
spin_double_bounce (spinners), 12
spin_dual_circle (spinners), 12
spin_dual_ring (spinners), 12
spin_ellipsis (spinners), 12
spin_facebook (spinners), 12
spin_fading_circles (spinners), 12
spin_fill (spinners), 12
spin_flower (spinners), 12
spin_flowers (spinners), 12
spin_folding_cube (spinners), 12
spin_gauge (spinners), 12
spin_google (spinners), 12
spin_half (spinners), 12
spin_heart (spinners), 12
spin_heartbeat (spinners), 12
spin_hexdots (spinners), 12
spin_hourglass (spinners), 12
spin_inner_circles (spinners), 12
spin_loader (spinners), 12
spin_loaders (spinners), 12
spin_orbit (spinners), 12
spin_orbiter (spinners), 12
spin_pixel (spinners), 12
spin_plus (spinners), 12
spin_pong (spinners), 12
spin_pulsar (spinners), 12
spin_pulse (spinners), 12
spin_pushing_shapes (spinners), 12
spin_puzzle (spinners), 12
spin_refresh (spinners), 12
spin_rhombus (spinners), 12
spin_ring (spinners), 12
spin_ripple (spinners), 12
spin_rotate (spinners), 12
spin_rotating_plane (spinners), 12
spin_seven_circle (spinners), 12
spin_solar (spinners), 12
spin_square_circle (spinners), 12
spin_squares (spinners), 12
spin_terminal (spinners), 12
spin_three_bounce (spinners), 12
spin_throbbing (spinners), 12
spin_throbber (spinners), 12
spin_timer (spinners), 12
spin_wandering_cubes (spinners), 12
spin_wave (spinners), 12
spin_whirly (spinners), 12
spin_wobblebar (spinners), 12
spinners, 12, 17, 19–21
transparent, 16
use_garcon (garcon), 2
use_hostess (hostess), 5
use_steward, 16
use_waiter (waiter), 17
use_waitress (waitress), 21
validate, 19
Waiter (waiterClass), 19
waiter, 16, 17
waiter_get_theme (waiterTheme), 21
waiter_hide (waiter), 17
waiter_hide_on_render (waiter), 17
waiter_on_busy (waiter), 17
waiter_preloader (waiter), 17
waiter_set_theme (waiterTheme), 21
waiter_show, 12
waiter_show (waiter), 17
waiter_show_on_load (waiter), 17
waiter_unset_theme (waiterTheme), 21
waiter_update (waiter), 17
waiter_use (waiter), 17
waiterClass, 19
waiterTheme, 21
Waitress (waitressClass), 22
waitress, 21
waitressClass, 22