

Package ‘freewall’

May 8, 2026

Title A Wrapper of the JavaScript Library 'Freewall'

Version 1.0.0

Description Creates dynamic grid layouts of images that can be included in 'Shiny' applications and R markdown documents.

License GPL-3

URL <https://github.com/stla/freewall>

BugReports <https://github.com/stla/freewall/issues>

Imports htmlwidgets, htmltools, jquerylib

Suggests shiny

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
Minh Nguyen [cph] (author of the JavaScript library 'Freewall')

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2024-04-30 09:00:05 UTC

Contents

alphabet	2
freewall	2
freewall-shiny	4
nature	5

Index	6
--------------	----------

alphabet	<i>Some images</i>
----------	--------------------

Description

Twenty-four urls pointing to an image.

Usage

```
alphabet()
```

Value

A character vector of 24 urls pointing to an image.

Examples

```
freewall(alphabet(), widths = 200, cellW = 200, cellH = "auto")
```

freewall	<i>The 'freewall' widget</i>
----------	------------------------------

Description

Creates a grid of images.

Usage

```
freewall(
  images,
  backgroundImages = FALSE,
  widths = 100,
  heights = NULL,
  width = "100%",
  draggable = FALSE,
  animate = TRUE,
  cellW = 100,
  cellH = 100,
  delay = 0,
  fixSize = NULL,
  gutterX = 10,
  gutterY = 10,
  keepOrder = TRUE,
  rightToLeft = FALSE,
  bottomToTop = FALSE,
  elementId = NULL
)
```

Arguments

images	character vector of paths or urls to some images; for a Shiny app, the image files must be located in the www subfolder
backgroundImages	Boolean, whether to render the images with the CSS property background-image or with img elements
widths	the widths of the images in pixels; if a single value is given, it will be used for all images
heights	the heights of the images in pixels; if a single value is given, it will be used for all images; setting the heights is necessary if backgroundImages=TRUE, otherwise this argument can be set to NULL
width	the width of the container
draggable	Boolean, whether to enable draggability
animate	Boolean, whether to animate
cellW	width of unit, a number of pixels or "auto"
cellH	height of unit, a number of pixels or "auto"
delay	time delay for showing an item
fixSize	see freewall options
gutterX	space between columns, a number of pixels or "auto"
gutterY	space between rows, a number of pixels or "auto"
keepOrder	Boolean, whether to keep the order of the images
rightToLeft	Boolean, whether to let the layout start render from right to left
bottomToTop	Boolean, whether to let the layout start render from bottom to top
elementId	a HTML id for the container (usually useless)

Value

A htmlwidget object.

Examples

```
freewall(  
  nature(), widths = 200, cellW = 200, cellH = 200, draggable = TRUE  
)
```

Description

Output and render functions for using `freewall` within Shiny applications and interactive Rmd documents.

Usage

```
freewallOutput(outputId, width = "100%", height = "auto")
```

```
renderFreewall(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

<code>outputId</code>	output variable to read from
<code>width, height</code>	a valid CSS dimension (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended
<code>expr</code>	an expression that generates a freewall
<code>env</code>	the environment in which to evaluate <code>expr</code>
<code>quoted</code>	logical, whether <code>expr</code> is a quoted expression (with <code>quote()</code>); this is useful if you want to save an expression in a variable

Value

`freewallOutput` returns an output element that can be included in a Shiny UI definition, and `renderFreewall` returns a `shiny.render.function` object that can be included in a Shiny server definition.

Examples

```
if(require(shiny) && interactive()) {
  library(shiny)
  library(freewall)

  ui <- fluidPage(
    freewallOutput("fw")
  )

  server <- function(input, output, session) {
    output$fw <- renderFreewall({
      freewall(alphabet(), widths = 200, cellW = 200, cellH = "auto")
    })
  }

  shinyApp(ui, server)
}
```

nature

Some images

Description

Nine urls pointing to an image.

Usage

```
nature()
```

Value

A character vector of nine urls pointing to an image.

Examples

```
freewall(  
  nature(), widths = 200, cellW = 200, cellH = "200", draggable = TRUE  
)
```

Index

alphabet, [2](#)

freewall, [2](#), [4](#)

freewall-shiny, [4](#)

freewallOutput (freewall-shiny), [4](#)

nature, [5](#)

renderFreewall (freewall-shiny), [4](#)