

Package: widgetframe (via r-universe)

September 12, 2024

Type Package

Title 'Htmlwidgets' in Responsive 'iframes'

Version 0.3.1

Description Provides two functions 'frameableWidget()', and 'frameWidget()'. The 'frameableWidget()' is used to add extra code to a 'htmlwidget' which allows it to be rendered correctly inside a responsive 'iframe'. The 'frameWidget()' is a 'htmlwidget' which displays content of another 'htmlwidget' inside a responsive 'iframe'. These functions allow for easier embedding of 'htmlwidgets' in content management systems such as 'wordpress', 'blogger' etc. They also allow for separation of widget content from main HTML content where CSS of the main HTML could interfere with the widget.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.1.0), htmlwidgets,

Imports htmltools, purrr, magrittr, utils, tools

RoxygenNote 6.0.1

URL <https://github.com/bhaskarvk/widgetframe>,
<https://bhaskarvk.github.io/widgetframe/>

BugReports <https://github.com/bhaskarvk/widgetframe/issues>

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository <https://trafficonese.r-universe.dev>

RemoteUrl <https://github.com/trafficonese/widgetframe>

RemoteRef HEAD

RemoteSha 8683d86ccdff351d5aacf13db6ec09b5ed9a1054

Contents

frameableWidget	2
frameOptions	3
frameWidget	3
saveWidgetframe	4
widgetframe	5
widgetframe-shiny	5
Index	6

frameableWidget	<i>Adds pymjs initialization code to a htmlwidget.</i>
-----------------	--

Description

This function augments a `htmlwidget` so that when saved, the resulting HTML document can be rendered correctly inside a responsive `iframe` (created using `Pym.js`) of another HTML document.

Usage

```
frameableWidget(widget, renderCallback = NULL)
```

Arguments

`widget` The widget to add the `pymjs` code to.

`renderCallback` An optional Javascript function wrapped in `JS()` which will be called when parent sends a resize event.

Details

Generate your `htmlwidget` in the normal way and then call this function passing in your widget. Then call `saveWidget()` and the saved HTML file is now embeddable inside a `Pym.js` `iframe` of another HTML document. See `Pym.js` documentation on how to create an HTML document with a responsive `iframe`.

See Also

[frameWidget\(\)](#).

Examples

```
## Not run:
library(leaflet)
l <- leaflet() %>% addTiles() %>% setView(0,0,1)
htmlwidgets::saveWidget(
  widgetframe::frameableWidget(l), 'some-directory-on-your-disk')

## End(Not run)
```

frameOptions	<i>Options for widget's iframe.</i>
--------------	-------------------------------------

Description

Taken from [Pym.js Documentation](#). In addition also check out the [iframe documentation](#).

Usage

```
frameOptions(xdomain = "*", title = NULL, name = NULL, id = NULL,
            allowfullscreen = FALSE, sandbox = NULL, lazyload = FALSE)
```

Arguments

xdomain	xdomain to validate messages received.
title	If passed it will be assigned to the iframe title attribute.
name	If passed it will be assigned to the iframe name attribute.
id	If passed it will be assigned to the iframe id attribute.
allowfullscreen	If TRUE it will set the iframe allowfullscreen attribute to true.
sandbox	If passed it will be assigned to the iframe sandbox attribute.
lazyload	If TRUE the child widget is lazy loaded using bLazy.js .

frameWidget	<i>A widget that wraps another widget inside a responsive iframe.</i>
-------------	---

Description

Uses [Pym.js](#). Pym.js embeds and resizes an iframe responsively (width and height) within its parent container. It also bypasses the usual cross-domain issues.

Usage

```
frameWidget(targetWidget, width = "100%", height = NULL,
            elementId = NULL, options = frameOptions())
```

Arguments

targetWidget	The widget to embed inside an iframe.
width	Defaults to 100 100, 200 (in pixel). This will override the width of the enclosed widget.
height	Defaults to NULL. You can either specify '10 100, 200 (in pixel). This will override the height of the enclosed widget.
elementId	The element ID of the parent widget.
options	Options for the iframe.

Details

This widget can be used in places where a HTML page's CSS rules or Javascript code can cause issues in a widget. Wrapping your widgets this way allows for the widget code to be unaffected by the parent HTML's CSS/JS. The target widget is conveniently displayed in a responsive iframe and not subject to parent HTML's CSS/JS.

See Also

[frameOptions\(\)](#).

Examples

```
## Not run:
l <- leaflet() %>% addTiles() %>% setView(0,0,1)
frameWidget(l)

## End(Not run)
```

saveWidgetframe

Save a widgetframe and its child widget to HTML files.

Description

Similar to [saveWidget\(\)](#) with the addition that both the parent widget and the enclosed child widget are saved to two different HTML files.

Usage

```
saveWidgetframe(widget, file, selfcontained = FALSE, libdir = NULL,
  background = "white", knitrOptions = list())
```

Arguments

widget	widgetframe to save
file	File to save the parent widget into. The child widget will be saved to 'base-name(file)_widget/index.html'.
selfcontained	Whether to save the parent and child HTMLs as a single self-contained files. WARNING: Setting this option to true will still result in two HTMLs, one for the parent and another for the child widget (with external resources base64 encoded), or files with external resources placed in an adjacent directory.
libdir	Directory to copy HTML dependencies into (defaults to filename_files).
background	Text string giving the html background color of the widget. Defaults to white.
knitrOptions	A list of knitr chunk options.

widgetframe	<i>widgetframe: A package for wrapping htmlwidgets in responsive iframes.</i>
-------------	---

Description

This package provides two functions `frameableWidget`, and `frameWidget`. The `frameableWidget` is used to add extra code to a `htmlwidget` which allows it to be rendered correctly inside a responsive `iframe`. The `frameWidget` is a `htmlwidget` which displays content of another `htmlwidget` inside a responsive `iframe`.

widgetframe-shiny	<i>Shiny bindings for widgetframe</i>
-------------------	---------------------------------------

Description

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

Usage

```
widgetframeOutput(outputId, width = "100%", height = "400px")
renderWidgetframe(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

<code>outputId</code>	output variable to read from
<code>width, height</code>	Must be a valid CSS unit (like <code>'100%'</code> , <code>'400px'</code> , <code>'auto'</code>) or a number, which will be coerced to a string and have <code>'px'</code> appended.
<code>expr</code>	An expression that generates a <code>widgetframe</code>
<code>env</code>	The environment in which to evaluate <code>expr</code> .
<code>quoted</code>	Is <code>expr</code> a quoted expression (with <code>quote()</code>)? This is useful if you want to save an expression in a variable.

Index

`frameableWidget`, [2](#), [5](#)

`frameOptions`, [3](#), [4](#)

`frameWidget`, [2](#), [3](#), [5](#)

JS, [2](#)

`renderWidgetframe (widgetframe-shiny)`, [5](#)

`saveWidget`, [2](#), [4](#)

`saveWidgetframe`, [4](#)

`widgetframe`, [5](#)

`widgetframe-package (widgetframe)`, [5](#)

`widgetframe-shiny`, [5](#)

`widgetframeOutput (widgetframe-shiny)`, [5](#)