

Package: r2d3 (via r-universe)

October 9, 2024

Type Package

Title Interface to 'D3' Visualizations

Version 0.2.5

Description Suite of tools for using 'D3', a library for producing dynamic, interactive data visualizations. Supports translating objects into 'D3' friendly data structures, rendering 'D3' scripts, publishing 'D3' visualizations, incorporating 'D3' in R Markdown, creating interactive 'D3' applications with Shiny, and distributing 'D3' based 'htmlwidgets' in R packages.

License BSD_3_clause + file LICENSE

Encoding UTF-8

Depends R (>= 3.1.2)

Imports htmlwidgets (>= 1.2), htmltools, jsonlite, rstudioapi

Suggests knitr, rmarkdown, R6, shiny, shinytest, testthat, webshot

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

URL <https://rstudio.github.io/r2d3/>, <https://github.com/rstudio/r2d3>

BugReports <https://github.com/rstudio/r2d3/issues>

VignetteBuilder knitr

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

RemoteUrl <https://github.com/rstudio/r2d3>

RemoteRef HEAD

RemoteSha becfb81989c7fabfe79dee2dde999190025d4ba3

Contents

as_d3_data	2
d3-shiny	2
html_dependencies_d3	3

r2d3	4
save_d3_html	5
save_d3_png	6

Index	8
--------------	----------

as_d3_data	<i>Convert object to D3 data</i>
------------	----------------------------------

Description

Generic method to transform R objects into D3 friendly data.

Usage

```
as_d3_data(x, ...)
```

```
## Default S3 method:
```

```
as_d3_data(x, ...)
```

Arguments

x	data
...	Additional arguments for generic methods

Details

The value returned from `as_d3_data()` should be one of:

- An R data frame. In this case the `HTMLWidgets.dataframeToD3()` JavaScript function will be called on the client to transform the data into D3 friendly (row-oriented) data; or
- A JSON object created using `jsonlite::toJSON`; or
- Any other R object which can be converted to JSON using `jsonlite::toJSON`.

d3-shiny	<i>Shiny bindings for d3</i>
----------	------------------------------

Description

Output and render functions for using d3 within Shiny applications and interactive Rmd documents.

Usage

```
d3Output(outputId, width = "100%", height = "400px")
```

```
renderD3(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
expr	An expression that generates a d3
env	The environment in which to evaluate expr.
quoted	Is expr a quoted expression (with quote())? This is useful if you want to save an expression in a variable.

html_dependencies_d3 *D3 HTML dependencies*

Description

Create HTML dependencies for D3 and optional extensions

Usage

```
html_dependencies_d3(version = c("6", "5", "4", "3"), extensions = NULL)
```

Arguments

version	Major version of D3
extensions	D3 extensions to include. Currently the only supported extension is "jetpack" (https://github.com/gka/d3-jetpack).

Details

Create list of HTML dependencies for D3. Each version has a distinct root D3 object so it's possible to combine multiple versions of D3 on a single page. For example, D3 v5 is accessed via d3v5 and D3 v4 is accessed via d3v4. Note however that D3 v3 is accessed via simply d3 (for compabililty with existing htmlwidgets that use this form).

Note

This function is exported for use by htmlwidgets. If you are using the r2d3() function to include D3 code within a document or application this dependency is included automatically so calling this function is unnecessary.

Examples

```
library(r2d3)
r2d3(
  data = c(0.3, 0.6, 0.8, 0.95, 0.40, 0.20),
  script = system.file("examples/barchart.js", package = "r2d3"),
  dependencies = "d3-jetpack"
)
```

r2d3

*D3 visualization***Description**

Visualize data using a custom D3 visualization script

Usage

```
r2d3(
  data,
  script,
  css = "auto",
  dependencies = NULL,
  options = NULL,
  d3_version = c("6", "5", "4", "3"),
  container = "svg",
  elementId = NULL,
  width = NULL,
  height = NULL,
  sizing = default_sizing(),
  viewer = c("internal", "external", "browser")
)
```

Arguments

data	Data to be passed to D3 script.
script	JavaScript file containing the D3 script.
css	CSS file containing styles. The default value "auto" will use any CSS file located alongside the script file with the same stem (e.g. "barplot.css" would be used for "barplot.js") as well as any CSS file with the name "styles.css".
dependencies	Additional HTML dependencies. These can take the form of paths to JavaScript or CSS files, or alternatively can be fully specified dependencies created with htmltools::htmlDependency .
options	Options to be passed to D3 script.
d3_version	Major D3 version to use, the latest minor version is automatically picked.
container	The 'HTML' container of the D3 output.
elementId	Use an explicit element ID for the widget (rather than an automatically generated one). Useful if you have other JavaScript that needs to explicitly discover and interact with a specific widget instance.
width	Desired width for output widget.
height	Desired height for output widget.
sizing	Widget sizing policy (see htmlwidgets::sizingPolicy).
viewer	"internal" to use the RStudio internal viewer pane for output; "external" to display in an external RStudio window; "browser" to display in an external browser.

Details

In order to scope CSS styles when multiple widgets are rendered, the Shadow DOM and the web-components polyfill is used, this feature can be turned off by setting the `r2d3.shadow` option to `FALSE`.

Examples

```
library(r2d3)
r2d3(
  data = c (0.3, 0.6, 0.8, 0.95, 0.40, 0.20),
  script = system.file("examples/barchart.js", package = "r2d3")
)
```

save_d3_html

Save a D3 visualization as HTML

Description

Save a D3 visualization to an HTML file (e.g. for sharing with others).

Usage

```
save_d3_html(
  d3,
  file,
  selfcontained = TRUE,
  libdir = NULL,
  background = "white",
  title = "D3 Visualization",
  knitrOptions = list()
)
```

Arguments

<code>d3</code>	D3 visualization to save
<code>file</code>	File to save HTML into
<code>selfcontained</code>	Whether to save the HTML as a single self-contained file (with external resources base64 encoded) or a file with external resources placed in an adjacent directory.
<code>libdir</code>	Directory to copy HTML dependencies into (defaults to <code>filename_files</code>).
<code>background</code>	Text string giving the html background color of the widget. Defaults to white.
<code>title</code>	Text to use as the title of the generated page.
<code>knitrOptions</code>	A list of knitr chunk options.

Details

Using selfcontained set to TRUE requires [pandoc](#) to be installed.

See Also

[save_d3_png\(\)](#)

Examples

```
library(r2d3)

viz <- r2d3(
  data = c(0.3, 0.6, 0.8, 0.95, 0.40, 0.20),
  script = system.file("examples/barchart.js", package = "r2d3")
)

save_d3_html(
  viz,
  file = tempfile(fileext = ".html"),
  selfcontained = FALSE
)
```

save_d3_png

Save a D3 visualization as a PNG image

Description

Save a D3 visualization to PNG (e.g. for including in another document).

Usage

```
save_d3_png(
  d3,
  file,
  background = "white",
  width = 992,
  height = 744,
  delay = 0.2,
  zoom = 1
)
```

Arguments

d3	D3 visualization to save
file	File to save HTML into
background	Text string giving the html background color of the widget. Defaults to white.
width	Image width

height	Image height
delay	Time to wait before taking screenshot, in seconds. Sometimes a longer delay is needed for all assets to display properly.
zoom	A number specifying the zoom factor. A zoom factor of 2 will result in twice as many pixels vertically and horizontally. Note that using 2 is not exactly the same as taking a screenshot on a HiDPI (Retina) device: it is like increasing the zoom to 200 doubling the height and width of the browser window. This differs from using a HiDPI device because some web pages load different, higher-resolution images when they know they will be displayed on a HiDPI device (but using zoom will not report that there is a HiDPI device).

Details

PNG versions of D3 visualizations are created by displaying them in an offscreen web browser and taking a screenshot of the rendered web page.

Using the `save_d3_png()` function requires that you install the **webshot** package, as well as the `phantom.js` headless browser (which you can install using the function `webshot::install_phantomjs()`).

See Also

[save_d3_html\(\)](#)

Index

[as_d3_data](#), [2](#)

[d3-shiny](#), [2](#)

[d3Output](#) ([d3-shiny](#)), [2](#)

[html_dependencies_d3](#), [3](#)

[htmltools::htmlDependency](#), [4](#)

[htmlwidgets::sizingPolicy](#), [4](#)

[jsonlite::toJSON](#), [2](#)

[r2d3](#), [4](#)

[renderD3](#) ([d3-shiny](#)), [2](#)

[save_d3_html](#), [5](#)

[save_d3_html\(\)](#), [7](#)

[save_d3_png](#), [6](#)

[save_d3_png\(\)](#), [6](#)