Package ‘fabricerin’

August 14, 2020

Title Create Easily Canvas in 'shiny' and 'RMarkdown' Documents

Version 0.1.2

Description Allows the user to implement easily canvas elements within a 'shiny' app or an 'RMarkdown' document.
   The user can create shapes, images and text elements within the canvas which can also be used as a drawing tool for taking notes.
   The package relies on the 'fabricjs' 'JavaScript' library. See <http://fabricjs.com/>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports htmltools, glue

URL https://github.com/feddelegrand7/fabricerin

BugReports https://github.com/feddelegrand7/fabricerin/issues

NeedsCompilation no

Author Mohamed El Fodil Ihaddaden [aut, cre],
   Garrick Aden-Buie [ctb],
   fabricjs contributors [ctb, cph] (fabricjs JavaScript library),
   jQuery contributors [ctb, cph] (jQuery JavaScript library),
   FileSaver.js contributors [ctb, cph] (FileSaver JavaScript library)

Maintainer Mohamed El Fodil Ihaddaden <ihaddaden.fo.deil@gmail.com>

Repository CRAN

Date/Publication 2020-08-14 17:20:07 UTC

R topics documented:

  fabric_curtail ......................................................... 2
  fabric_drawing ....................................................... 3
  fabric_image .......................................................... 4
  fabric_image_add ..................................................... 6


**fabric_curtail**

Add a background or an overlay image to a preexisting canvas

**Description**

Add a background or an overlay image to a preexisting canvas

**Usage**

```
fabric_curtail(cid, imgsrc, type = "background")
```

**Arguments**

- **cid**: the id of the canvas element
- **imgsrc**: the URL source of the image
- **type**: whether to use an image as a 'background' or as an 'overlay'

**Value**

A canvas with a background or overlay image

**Examples**

```r
if (interactive()) {


  ui <- fluidPage(

    fabric_shape(cid = "canvas123",
                  shapeId = "tri1",
                  shape = "Triangle",
                  fill = "darkblue"),

    fabric_curtail(cid = "canvas123",
                   imgsrc = img,
                   type = "background"

  )

)```

Create a canvas element for drawing

Description

Create a canvas element for drawing

Usage

fabric_drawing(
  cid,
  cwidth = 800,
  cheight = 600,
  cfill = "#FFFFFF",
  drawingWidth = 2,
  gumSize = 10
)

Arguments

- `cid` the id of the canvas element
- `cwidth` the width of the canvas element
- `cheight` the height of the canvas element
- `cfill` the color of the canvas element. Default to FFFFFF (white)
- `drawingWidth` the width of the drawing output. Default to 2
- `gumSize` specify the size of the gum. Defaults to 10

Value

an HTML canvas element
Examples

```r
if (interactive()) {
  ui <- fluidPage(
    h1("Draw some stuff here"),
    fabric_drawing(cid = "canvas1")
  )
  server <- function(input, output) {}  
  shinyApp(ui = ui, server = server)
}
```

---

**fabric_image**  
*Insert external images inside canvas element*

**Description**

Insert external images inside canvas element

**Usage**

```r
fabric_image(
  cid,  
  cwidth = 800,  
  cheight = 600,  
  cfill = "#FFFFFF",  
  imgId,  
  imgsrc,  
  imgwidth = 500,  
  imgheight = 500,  
  left = 100,  
  top = 100,  
  angle = 0,  
  opacity = 1,  
  strokecolor = "darkblue",  
  strokewidth = 1,  
  selectable = TRUE,  
  isDrawingMode = FALSE
)
```
**Arguments**

- **cid**  
  the id of the canvas element
- **cwidth**  
  the width of the canvas element. Defaults to 800
- **cheight**  
  the height of the canvas element. Defaults to 600
- **cfill**  
  the color of the canvas element
- **imgId**  
  the id of the image
- **imgsnc**  
  the URL source of the image
- **imgwidth**  
  the width of the image. Defaults to 500
- **imgheight**  
  the height of the image. Defaults to 500
- **left**  
  the image’s position from the left relative to the canvas element. Defaults to 100
- **top**  
  the image’s position from the top relative to the canvas element. Defaults to 100
- **angle**  
  the angle of rotation of the image. Defaults to 0 (no rotation)
- **opacity**  
  the opacity of the image (from 0 to 1). Defaults to 1
- **strokecolor**  
  the stroke color of the image. Defaults to ‘darkblue’
- **strokewidth**  
  the stroke width of the image. Defaults to 1
- **selectable**  
  logical. If TRUE, the user can modify interactively the image’s size, position and rotation. Defaults to TRUE
- **isDrawingMode**  
  logical. If TRUE, the user can draw inside the canvas element.

**Value**

an image inside a canvas element

**Examples**

```r
if (interactive()) {

  img <- "https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/R_logo.svg/724px-R_logo.svg.png"

  ui <- fluidPage(

    fabric_image(cid = "cimage",
      cfill = "lightblue",
      imgId = "Rimg",
      imgsrc = img)

  )

  server <- function(input, output) {}

  shinyApp(ui = ui, server = server)
```
fabric_image_add

Add an image to a preexisting canvas element

Description

Add an image to a preexisting canvas element

Usage

```
fabric_image_add(
  cid,
  imgId,
  imgsnc,
  imgwidth = 500,
  imgheight = 500,
  left = 100,
  top = 100,
  angle = 0,
  opacity = 1,
  strokecolor = "darkblue",
  strokewidth = 1,
  selectable = TRUE
)
```

Arguments

- `cid`: the id of the canvas element you want to add your image to
- `imgId`: the of the image
- `imgsnc`: the URL source of the image
- `imgwidth`: the width of the image. Defaults to 500
- `imgheight`: the height of the image. Defaults to 500
- `left`: the image’s position from the left relative to the canvas element. Defaults to 100
- `top`: the image’s position from the top relative to the canvas element. Defaults to 100
- `angle`: the angle of rotation of the image. Defaults to 0 (no rotation)
- `opacity`: the opacity of the image (from 0 to 1). Defaults to 1
- `strokecolor`: the stroke color of the image. Defaults to 'darkblue'
- `strokewidth`: the stroke width of the image. Defaults to 1
- `selectable`: logical. If TRUE, the user can modify interactively the image’s size, position and rotation. Defaults to TRUE
fabric_shape

Value
an image inside a preexisting canvas element

Examples

```r
if (interactive()) {

  img1 <- "https://upload.wikimedia.org/wikipedia/commons/thumb/1/1b/R_logo.svg/724px-R_logo.svg.png"
  img2 <- "https://raw.githubusercontent.com/rstudio/hex-stickers/master/PNG/dplyr.png"
  ui <- fluidPage(

    fabric_image(cid = "cimage",
                  imgId = "Rimg",
                  imgsrc = img1,
                  imgheight = 200,
                  imgwidth = 200),

    fabric_image_add(cid = "cimage",
                      imgId = "rstudioimg",
                      imgsrc = img2,
                      imgwidth = 200,
                      imgheight = 200,
                      left = 400)

  server <- function(input, output) {}

  shinyApp(ui = ui, server = server)
  }
```

fabric_shape Create shapes inside a canvas

Description
Create shapes inside a canvas

Usage

```r
fabric_shape(
  cid,
  cwidth = 800,
  cheight = 600,
  cfill = "#FFFFFF",
  shapeId,
  shape = "Rect",
  left = 100,
```
top = 100, 
fill = "red", 
width = 200, 
height = 200, 
angle = 0, 
opacity = 1, 
strokecolor = "darkblue", 
strokewidth = 5, 
selectable = TRUE, 
isDrawingMode = FALSE, 
radius = NULL, 
xPolygon = NULL, 
yPolygon = NULL
)

Arguments

cid the id of the canvas element
cwidth the width of the canvas element. Defaults to 800
cheight the height of the canvas element. Defaults to 600
cfill the color of the canvas element
shapeId the id of the shape object
shape the shape of the object. Choices include 'Circle', 'Triangle' and 'Rect'. Defaults to 'Rect'
left the shape’s position from the left relative to the canvas element. Defaults to 100
top the shape’s position from the top relative to the canvas element. Defaults to 100
cfill the color of the shape. Defaults to 'red'
cwidth the width of the shape. Defaults to 200
cheight the height of the shape. Defaults to 200
cangle the angle of rotation of the shape. Defaults to 0 (no rotation)
copacity the opacity of the shape (from 0 to 1). Defaults to 1
cstrokecolor the stroke color of the shape. Defaults to 'darkblue'
cstrokewidth the stroke width of the shape. Defaults to 5.
cselectable logical. If TRUE, the user can modify interactively the shape’s size, position and rotation. Defaults to TRUE
cisDrawingMode logical. If TRUE, the user can draw inside the canvas element.
cradius mandatory if the chosen shape is a 'Circle'. Defaults to NULL
cxPolygon a vector of the coordinate points of the polygon, from the left.
cyPolygon a vector of the coordinate points of the polygon, from the top

Value

a shape object inside a canvas
**Examples**

```r
if(interactive()){
  ui <- fluidPage(
    h2("Below you'll find a red Rectangle with a darkblue stroke"),
    fabric_shape(cid = "canvas", shapeId = "shape1", shape = "Rect")
  )
  server <- function(input, output) {
    
  }
  shinyApp(ui = ui, server = server)
}
```

**fabric_shape_add**  
*Add a shape object to a preexisting canvas element*

**Description**

Add a shape object to a preexisting canvas element

**Usage**

```r
fabric_shape_add(cid, shapeId, shape = "Rect", left = "100",
  top = "100", fill = "red", width = 200,
  height = 200, angle = 0, opacity = 1,
  strokecolor = "darkblue", strokewidth = 5,
  selectable = TRUE, radius = NULL,)
```
Arguments

- `cid` : the id of the canvas element you want to add your shape to
- `shapeId` : the id of the shape object
- `shape` : the shape of the object. Choices include 'Circle', 'Triangle' and 'Rect'. Defaults to 'Rect'
- `left` : the shape's position from the left relative to the canvas element. Defaults to 100
- `top` : the shape's position from the top relative to the canvas element. Defaults to 100
- `fill` : the color of the shape. Defaults to 'red'
- `width` : the width of the shape. Defaults to 200
- `height` : the height of the shape. Defaults to 200
- `angle` : the angle of rotation of the shape. Defaults to 0 (no rotation)
- `opacity` : the opacity of the shape. Defaults to 1
- `strokecolor` : the stroke color of the shape. Defaults to 'darkblue'
- `strokewidth` : the stroke width of the shape. Defaults to 5.
- `selectable` : logical. If TRUE, the user can modify interactively the shape. Defaults to TRUE
- `radius` : Mandatory if the chosen shape is a 'Circle'. Defaults to NULL
- `xPolygon` : a vector of the coordinate points of the polygon, from the left.
- `yPolygon` : a vector of the coordinate points of the polygon, from the top

Value

- a shape object inside a preexisting canvas element

Examples

```r
if (interactive()) {
  ui <- fluidPage(
    fabric_shape(cid = "canvas",
                  shapeId = "shape1",
                  shape = "Rect",
                  left = 130,
                  top = 200),
    fabric_shape_add(cid = "canvas",
                     shapeId = "shapo",
                     shape = "Circle",
                     radius = 30),
   "
```

fabric_text

```
left = 100,
top = 100),

fabric_shape_add(cid = "canvas",
    shapeId = "shapa",
    shape = "Circle",
    radius = 30,
    left = 200,
    top = 100),

fabric_shape_add(cid = "canvas",
    shapeId = "shapox",
    shape = "Circle",
    radius = 30,
    left = 300,
    top = 100),

fabric_shape_add(cid = "canvas",
    shapeId = "shapor",
    shape = "Circle",
    radius = 30,
    left = 300,
    top = 100)
```

server <- function(input, output) {}

shinyApp(ui = ui, server = server)
```

fabric_text

Insert text within canvas element

Description

Insert text within canvas element

Usage

fabric_text(
    cid,
    cwidth = 800,
    cheight = 600,
    cfill = "#FFFFFF",
    textId,
    text,
    left = 100,
top = 100,
fill = "#2F3941",
angle = 0,
opacity = 1,
fontFamily = "Comic Sans",
fontSize = 40,
fontStyle = "normal",
strokecolor = "#282A36",
strokewidth = 1,
fontWeight = "normal",
underline = FALSE,
linethrough = FALSE,
overline = FALSE,
selectable = TRUE,
shadow = FALSE,
shadowCol = "#FFFAF0",
textAlign = "center",
lineHeight = 1,
textBackgroundColor = NULL,
isDrawingMode = FALSE
)

Arguments

cid the id of the canvas element
cwidth the width of the canvas element. Defaults to 800
cheight the height of the canvas element. Defaults to 600
cfill the color of the canvas element
textId the id of the text
text the content of the text
left the text's position from the left relative to the canvas element. Defaults to 100
top the text's position from the top relative to the canvas element. Defaults to 100
fill the text's color. Defaults to "#2F3941" (dark shade of cyan-blue)
angle the angle of rotation of the text. Defaults to 0 (no rotation)
opacity text opacity (from 0 to 1). Defaults to 1
fontFamily the font family of the text. Defaults to 'Comic Sans'
fontSize text sizing. Defaults to 40
fontStyle the font style of the text. Either 'normal' or 'italic'
strokecolor the stroke color of the text Defaults to "#282A36" (Very dark grayish blue)
strokewidth the stroke width of the text. Defaults to 1
fontWeight allows the user to make text thicker or thinner. Keywords can be used ('normal', 'bold'), or numbers. Defaults to 'normal'
underline logical. Whether to underline the text or not. Defaults to FALSE
linethrough logical. Whether to insert a line through the text or not. Defaults to FALSE
overline logical. Whether to put a line above the text or not. Defaults to FALSE
selectable logical. If TRUE, the user can modify interactively the image’s size, position and rotation. Defaults to TRUE
shadow logical. If TRUE a text shadow will be inserted behind the raw text. Defaults to FALSE
shadowCol the color of the text shadow. Defaults to #FFFAF0 (floral white)
textAlign the alignment of text. Useful when there are line breaks. Defaults to "center"
lineHeight the height of the line breaks. Defaults to 1
textBackgroundColor the background color of the text, defaults to NULL
isDrawingMode logical. If TRUE, the user can draw inside the canvas element.

Value

a text object within a canvas element

Examples

if (interactive()) {

ui <- fluidPage(

fabric_text(cid = "can",
    textId = "text",
    text = "But A Hero Is A Guy Who Gives Out The Meat To Everyone Else.",
    cfill = "#DD5347",
    left = 120,
    shadowCol = "blue",
    fontSize = 20,
    fontWeight = "bold",
    lineHeight = 3
)
)
server <- function(input, output) {}

shinyApp(ui = ui, server = server)
}
Add text within preexisting canvas element

**Description**

Add text within preexisting canvas element

**Usage**

```javascript
fabric_text_add(
    cid,
    textId,
    text,
    left = 100,
    top = 100,
    fill = "#2F3941",
    angle = 0,
    opacity = 1,
    fontFamily = "Comic Sans",
    fontSize = 40,
    fontStyle = "normal",
    strokecolor = "#282A36",
    strokewidth = 1,
    fontWeight = "normal",
    underline = FALSE,
    linethrough = FALSE,
    overline = FALSE,
    selectable = TRUE,
    shadow = FALSE,
    shadowCol = "#324C63",
    textAlign = "center",
    lineHeight = 1,
    textBackgroundColor = NULL
)
```

**Arguments**

- **cid**: the id of the canvas element
- **textId**: the id of the text
- **text**: the content of the text
- **left**: the text’s position from the left relative to the canvas element. Defaults to 100
- **top**: the text’s position from the top relative to the canvas element. Defaults to 100
- **fill**: the text’s color. Defaults to ‘#2F3941’ (dark shade of cyan-blue)
- **angle**: the angle of rotation of the text. Defaults to 0 (no rotation)
- **opacity**: text opacity (from 0 to 1). Defaults to 1
fontFamily the font family of the text. Defaults to 'Comic Sans'
fontSize text sizing. Defaults to 40
fontStyle the font style of the text. Either 'normal' or 'italic'
strokecolor the stroke color of the text. Defaults to '#282A36' (Very dark grayish blue)
strokewidth the stroke width of the text. Defaults to 1
fontWeight allows the user to make text thicker or thinner. Keywords can be used ('normal', 'bold'), or numbers. Defaults to 'normal'
underline logical. Whether to underline the text or not. Defaults to FALSE
linethrough logical. Whether to insert a line through the text or not. Defaults to FALSE
overline logical. Whether to put a line above the text or not. Defaults to FALSE
selectable logical. If TRUE, the user can modify interactively the image’s size, position and rotation. Defaults to TRUE
shadow logical. If TRUE a text shadow will be inserted behind the raw text. Defaults to FALSE
shadowCol the color of the text shadow. Defaults to #FFFAF0 (floral white)
textAlign the alignment of text. Useful when there are line breaks. Defaults to "center"
lineHeight the height of the line breaks. Defaults to 1
textBackgroundColor the background color of the text, defaults to NULL

Value

a text object within a preexisting canvas element

Examples

if (interactive()) {
  ui <- fluidPage(

    fabric_shape(cid = "canvas123",
                 cfill = "lightblue",
                 cwidth = 1000,
                 shapeId = "tri1",
                 shape = "Triangle",
                 fill = "darkblue"),

    fabric_text_add(cid = "canvas123",
                    textId = "txt1",
                    text = "This is a darkblue Triangle!",
                    left = 350)
  )
}
server <- function(input, output) {}
shinyApp(ui = ui, server = server)
}

filesaver_dependency  
Create an HTML dependency for FileSaver.js

Description
Create an HTML dependency for FileSaver.js
Create an HTML dependency for fabric.js
Create an HTML dependency for jQuery

Usage
filesaver_dependency()
fabric_dependency()
jquery_dependency()
Index

fabric_curtail, 2
fabric_dependency
   (filesaver_dependency). 16
fabric_drawing, 3
fabric_image, 4
fabric_image_add, 6
fabric_shape, 7
fabric_shape_add, 9
fabric_text, 11
fabric_text_add, 14
filesaver_dependency, 16

jquery_dependency
   (filesaver_dependency). 16