Package 'httpgd'

January 30, 2023

Type Package

Title A 'HTTP' Server Graphics Device

Version 1.3.1

Description A graphics device for R that is accessible via network protocols.
This package was created to make it easier to embed live R graphics in
integrated development environments and other applications.
The included 'HTML/JavaScript' client (plot viewer) aims to provide a better overall user experience when dealing with R graphics.
The device asynchronously serves graphics via 'HTTP' and 'WebSockets'.

License GPL (>= 2)

Depends R (>= 3.2.0)

Imports later (>= 1.1.0), systemfonts (>= 1.0.0)

LinkingTo cpp11 (>= 0.2.4), BH (>= 1.75.0), later, systemfonts

Suggests testthat, xml2 (>= 1.0.0), fontquiver (>= 0.2.0), knitr,
rmarkdown

RoxygenNote 7.2.3

Encoding UTF-8

SystemRequirements C++17, libpng, cairo, freetype2, fontconfig

URL https://github.com/nx10/httpgd, https://nx10.github.io/httpgd/

BugReports https://github.com/nx10/httpgd/issues

VignetteBuilder knitr

NeedsCompilation yes

Author Florian Rupprecht [aut, cre] (<https://orcid.org/0000-0002-1795-8624>),
Kun Ren [ctb],
Tatsuya Shima [ctb],
Jeroen Ooms [ctb] (<https://orcid.org/0000-0002-4035-0289>),
Hadley Wickham [cph] (Author of included svglite code),
Lionel Henry [cph] (Author of included svglite code),
Thomas Lin Pedersen [cph] (Author and creator of included svglite code),
T Jake Luciani [cph] (Author of included svglite code),

Matthieu Decorde [cph] (Author of included svglite code),
Vaudor Lise [cph] (Author of included svglite code),
Tony Plate [cph] (Contributor to included svglite code),
David Gohel [cph] (Contributor to included svglite code),
Yixuan Qiu [cph] (Contributor to included svglite code),
Håkon Malmedal [cph] (Contributor to included svglite code),
RStudio [cph] (Copyright holder of included svglite code),
Brett Robinson [cph] (Author of included belle library),
Google [cph] (Copyright holder of included material design icons),
Victor Zverovich [cph] (Author of included fmt library)

Maintainer Florian Rupprecht <floruppr@gmail.com>

Repository CRAN

Date/Publication 2023-01-30 14:00:02 UTC

R topics documented:

`httpgd-package` *httpgd: HTTP server graphics device*

Description

Asynchronous HTTP server graphics device.

hgd *Asynchronous HTTP server graphics device.*

Description

This function initializes a httpgd graphics device and starts a local webserver, that allows for access via HTTP and WebSockets. A link will be printed by which the web client can be accessed using a browser.

Usage

```
hgd(
  host = getOption("httpgd.host", "127.0.0.1"),
  port = getOption("httpgd.port", 0),
  width = getOption("httpgd.width", 720),
  height = getOption("httpgd.height", 576),
  bg = getOption("httpgd.bg", "white"),
  pointsize = getOption("httpgd.pointsize", 12),
  system_fonts = getOption("httpgd.system_fonts", list()),
  user_fonts = getOption("httpgd.user_fonts", list()),
  cors = getOption("httpgd.cors", FALSE),
  token = getOption("httpgd.token", TRUE),
  silent = getOption("httpgd.silent", FALSE),
  websockets = getOption("httpgd.websockets", TRUE),
  webserver = getOption("httpgd.webserver", TRUE),
  fix_text_width = getOption("httpgd.fix_text_width", TRUE),
  extra_css = getOption("httpgd.extra_css", "")
)
```

Arguments

`host`	Server hostname. Set to `"0.0.0.0"` to enable remote access. We recommend to **only enable remote access in trusted networks**. The network security of httpgd has not yet been properly tested.
`port`	Server port. If this is set to `0`, an open port will be assigned.
`width`	Graphics device width (pixels).
`height`	Graphics device height (pixels).
`bg`	Background color.
`pointsize`	Graphics device point size.
`system_fonts`	Named list of font names to be aliased with fonts installed on your system. If unspecified, the R default families `sans`, `serif`, `mono` and `symbol` are aliased to the family returned by `systemfonts::font_info()`.
`user_fonts`	Named list of fonts to be aliased with font files provided by the user rather than fonts properly installed on the system. The aliases can be fonts from the fontquiver package, strings containing a path to a font file, or a list containing

	name and file elements with name indicating the font alias in the SVG output and file the path to a font file.
cors	Toggles Cross-Origin Resource Sharing (CORS) header. When set to TRUE, CORS header will be set to "*".
token	(Optional) security token. When set, all requests need to include a token to be allowed. (Either in a request header (X-HTTPGD-TOKEN) field or as a query parameter.) This parameter can be set to TRUE to generate a random 8 character alphanumeric token. A random token of the specified length is generated when it is set to a number. FALSE deactivates the token.
silent	When set to FALSE no information will be printed to console.
websockets	Use websockets.
webserver	Can be set to FALSE for offline mode. In offline mode the device is only accessible via R.
fix_text_width	Should the width of strings be fixed so that it doesn't change between SVG renderers depending on their font rendering? Defaults to TRUE. If TRUE each string will have the textLength CSS property set to the width calculated by systemfonts and lengthAdjust='spacingAndGlyphs'. Setting this to FALSE can be beneficial for heavy post-processing that may change content or style of strings, but may lead to inconsistencies between strings and graphic elements that depend on the dimensions of the string (e.g. label borders and background).
extra_css	Extra CSS to be added to the SVG. This can be used to embed webfonts.

Details

All font settings and descriptions are adopted from the excellent 'svglite' package.

Value

No return value, called to initialize graphics device.

Examples

```
if (interactive()) {
  hgd() # Initialize graphics device and start server
  hgd_browse() # Or copy the displayed link in the browser

  # Plot something
  x <- seq(0, 3 * pi, by = 0.1)
  plot(x, sin(x), type = "l")

  dev.off() # alternatively: hgd_close()
}
```

hgd_browse *Open httpgd URL in the browser.*

Description

This function will only work after starting a device with `hgd()`.

Usage

```
hgd_browse(..., which = dev.cur(), browser = getOption("browser"))
```

Arguments

`...`	Parameters passed to `hgd_url()`.
`which`	Which device (ID).
`browser`	Program to be used as HTML browser.

Value

URL.

Examples

```
if (interactive()) {
  hgd() # Initialize graphics device and start server
  hgd_browse() # Or copy the displayed link in the browser

  # Plot something
  x <- seq(0, 3 * pi, by = 0.1)
  plot(x, sin(x), type = "l")

  dev.off() # alternatively: hgd_close()
}
```

hgd_clear *Clear all httpgd plot pages.*

Description

This function will only work after starting a device with `hgd()`.

Usage

```
hgd_clear(which = dev.cur())
```

Arguments

which	Which device (ID).

Value

Whether there were any pages to remove.

Examples

```
hgd()

plot(1, 1)
hgd_clear()
hgd_clear()

dev.off()
```

---

hgd_close *Close httpgd device.*

Description

This achieves the same effect as grDevices::dev.off(), but will only close the device if it has the httpgd type.

Usage

```
hgd_close(which = dev.cur(), all = FALSE)
```

Arguments

which	Which device (ID).
all	Should all running httpgd devices be closed.

Value

Number and name of the new active device (after the specified device has been shut down).

Examples

```
hgd()
hist(rnorm(100))
hgd_close() # Equvalent to dev.off()

hgd()
hgd()
hgd()
hgd_close(all = TRUE)
```

hgd_generate_token *Generate random alphanumeric token.*

Description

This is mainly used internally by httpgd, but exposed for testing purposes.

Usage

```
hgd_generate_token(len)
```

Arguments

len	Token length (number of characters).

Value

Random token string.

Examples

```
hgd_generate_token(6)
```

hgd_id *Query httpgd plot IDs*

Description

Query httpgd graphics device static plot IDs. Available plot IDs starting from `index` will be returned. `limit` specifies the number of plots. This function will only work after starting a device with `hgd()`.

Usage

```
hgd_id(index = 0, limit = 1, which = dev.cur(), state = FALSE)
```

Arguments

index	Plot index. If this is set to `0`, the last page will be selected.
limit	Limit the number of returned IDs. If this is set to a value > 1 the returned type is a list if IDs.
which	Which device (ID).
state	Include the current device state in the returned result (see also: `hgd_state()`).

Value

a list contains plot IDs.

Examples

```
hgd()

plot.new()
text(.5, .5, "#1")
plot.new()
text(.5, .5, "#2")
plot.new()
text(.5, .5, "#3")
hgd_id() # The last one
hgd_id(2) # The second one
hgd_id(1, limit = Inf) # The first one and all the followings

dev.off()
```

---

hgd_info *httpgd device information.*

Description

Access general information of a httpgd graphics device. This function will only work after starting a device with `hgd()`.

Usage

```
hgd_info(which = dev.cur())
```

Arguments

`which`	Which device (ID).

Value

List of status variables with the following named items: `$id`: Server unique ID, `$version`: httpgd and library versions.

Examples

```
hgd()

hgd_info()

dev.off()
```

hgd_inline *Inline SVG rendering.*

Description

Convenience function for quick inline SVG rendering. This is similar to `hgd_svg()` but the plotting code is specified inline and an offline httpgd graphics device is managed (created and closed) automatically. Starting a device with `hgd()` is therefore not necessary.

Usage

```
hgd_inline(
  code,
  page = 0,
  page_width = -1,
  page_height = -1,
  zoom = 1,
  renderer = "svg",
  file = NA,
  ...
)
```

Arguments

`code`	Plotting code. See examples for more information.
`page`	Plot page to render. If this is set to `0`, the last page will be selected. Can be set to a numeric plot index or plot ID (see `hgd_id()`).
`page_width`	Width of the plot. If this is set to `-1`, the last width will be selected.
`page_height`	Height of the plot. If this is set to `-1`, the last height will be selected.
`zoom`	Zoom level. (For example: `2` corresponds to 200%, `0.5` would be 50%.)
`renderer`	Renderer.
`file`	Filepath to save SVG. (No file will be created if this is `NA`)
`...`	Additional parameters passed to `hgd(webserver=FALSE, ...)`

Value

Rendered SVG string.

Examples

```
hgd_inline({
  hist(rnorm(100))
})

s <- hgd_inline({
  plot.new()
```

```
  lines(c(0.5, 1, 0.5), c(0.5, 1, 1))
})
cat(s)
```

`hgd_plot` *Render httpgd plot.*

Description

This function will only work after starting a device with `hgd()`.

Usage

```
hgd_plot(
  page = 0,
  width = -1,
  height = -1,
  zoom = 1,
  renderer = "svg",
  which = dev.cur(),
  file = NA
)
```

Arguments

`page`	Plot page to render. If this is set to `0`, the last page will be selected. Can be set to a numeric plot index or plot ID (see `hgd_id()`).
`width`	Width of the plot. If this is set to `-1`, the last width will be selected.
`height`	Height of the plot. If this is set to `-1`, the last height will be selected.
`zoom`	Zoom level. (For example: `2` corresponds to 200%, `0.5` would be 50%.)
`renderer`	Renderer.
`which`	Which device (ID).
`file`	Filepath to save SVG. (No file will be created if this is NA)

Value

Rendered SVG string.

Examples

```
hgd()

plot(1, 1)
s <- hgd_plot(width = 600, height = 400)

hist(rnorm(100))
```

```
tf <- tempfile()
on.exit(unlink(tf))
hgd_plot(file = tf, width = 600, height = 400)

dev.off()
```

---

hgd_remove *Remove a httpgd plot page.*

Description

This function will only work after starting a device with `hgd()`.

Usage

```
hgd_remove(page = 0, which = dev.cur())
```

Arguments

page	Plot page to remove. If this is set to `0`, the last page will be selected. Can be set to a numeric plot index or plot ID (see `hgd_id()`).
which	Which device (ID).

Value

Whether the page existed (and thereby was successfully removed).

Examples

```
hgd()

plot(1, 1) # page 1
hist(rnorm(100)) # page 2
hgd_remove(page = 1) # remove page 1

dev.off()
```

hgd_renderers *httpgd device renderers.*

Description

Get a list of available renderers. This function will only work after starting a device with `hgd()`.

Usage

```
hgd_renderers(which = dev.cur())
```

Arguments

`which`	Which device (ID).

Value

List of renderers with the following named items: `$id`: Renderer ID, `$mime`: File mime type, `$ext`: File extension, `$name`: Human readable name, `$type`: Renderer type (currently either `plot` or `other`), `$bin`: Is the file a binary blob or text.

Examples

```
hgd()

hgd_renderers()

dev.off()
```

hgd_state *httpgd device status.*

Description

Access status information of a httpgd graphics device. This function will only work after starting a device with `hgd()`.

Usage

```
hgd_state(which = dev.cur())
```

Arguments

`which`	Which device (ID).

Value

List of status variables with the following named items: `$host`: Server hostname, `$port`: Server port, `$token`: Security token, `$hsize`: Plot history size (how many plots are accessible), `$upid`: Update ID (changes when the device has received new information), `$active`: Is the device the currently activated device.

Examples

```
hgd()

hgd_state()
plot(1, 1)
hgd_state()

dev.off()
```

`hgd_svg` *Render httpgd plot to SVG.*

Description

This function will only work after starting a device with `hgd()`.

Usage

```
hgd_svg(
  page = 0,
  width = -1,
  height = -1,
  zoom = 1,
  which = dev.cur(),
  file = NA
)
```

Arguments

`page`	Plot page to render. If this is set to `0`, the last page will be selected. Can be set to a numeric plot index or plot ID (see `hgd_id()`).
`width`	Width of the plot. If this is set to `-1`, the last width will be selected.
`height`	Height of the plot. If this is set to `-1`, the last height will be selected.
`zoom`	Zoom level. (For example: `2` corresponds to 200%, `0.5` would be 50%.)
`which`	Which device (ID).
`file`	Filepath to save SVG. (No file will be created if this is NA)

Value

Rendered SVG string.

Examples

```
hgd()

plot(1, 1)
s <- hgd_svg(width = 600, height = 400)

hist(rnorm(100))
tf <- tempfile()
on.exit(unlink(tf))
hgd_svg(file = tf, width = 600, height = 400)

dev.off()
```

`hgd_test_pattern` *Plot a test pattern that can be used to evaluate and compare graphics devices.*

Description

Plot a test pattern that can be used to evaluate and compare graphics devices.

Usage

```
hgd_test_pattern()
```

Value

No return value.

Examples

```
hgd_test_pattern()
```

`hgd_url` *httpgd URL.*

Description

Generate URLs to the plot viewer or to plot SVGs. This function will only work after starting a device with `hgd()`.

Usage

```
hgd_url(
  endpoint = "live",
  which = dev.cur(),
  websockets = TRUE,
  width = -1,
  height = -1,
  renderer = NA,
  history = TRUE,
  host = NULL,
  port = NULL,
  explicit = FALSE
)
```

Arguments

endpoint	API endpoint. The default, "live" is the HTML/JS plot viewer. Can be set to a numeric plot index or plot ID (see hgd_id()) to obtain the direct URL to the SVG.
which	Which device (ID).
websockets	Use websockets.
width	Width of the plot. (Only used when endpoint is "svg", or a plot index or ID.)
height	Height of the plot. (Only used when endpoint is "svg", or a plot index or ID.)
renderer	Renderer.
history	Should the plot history sidebar be visible.
host	Replaces hostname.
port	Replaces port.
explicit	Ads hgd={host}:{port} query parameter. Needed for host resolution in some editors.

Value

URL.

Examples

```
hgd()

plot(1, 1)
hgd_url(0)
hgd_url(hgd_id(), width = 800, height = 600)

dev.off()
```

`hgd_view` *Open httpgd URL in the IDE.*

Description

Global option `viewer` needs to be set to a function that accepts the client URL as a parameter.

Usage

```
hgd_view()
```

Details

This function will only work after starting a device with `hgd()`.

Value

`viewer` function return value.

Examples

```
if (interactive() && !is.null(getOption("viewer"))) {
  hgd()

  hgd_view()
  hist(rnorm(100))

  dev.off()
}
```

Index