Package 'webshot2'

July 22, 2025

Title Take Screenshots of Web Pages

Version 0.1.2

Description Takes screenshots of web pages, including Shiny applications and R Markdown documents. 'webshot2' uses headless Chrome or Chromium as the browser back-end.

License MIT + file LICENSE

URL https://rstudio.github.io/webshot2/, https://github.com/rstudio/webshot2

BugReports https://github.com/rstudio/webshot2/issues

Depends R (>= 3.2)

Imports callr, chromote (>= 0.1.0), later, magrittr, promises

Suggests httpuv, rmarkdown, shiny

Config/Needs/website r-lib/pkgdown, rstudio/bslib

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Winston Chang [aut, cre], Barret Schloerke [ctb] (ORCID: <https://orcid.org/0000-0001-9986-114X>), Posit Software, PBC [cph, fnd] (ROR: <https://ror.org/03wc8by49>)

Maintainer Winston Chang <winston@posit.co>

Repository CRAN

Date/Publication 2025-04-23 21:40:02 UTC

Contents

---

appshot *Take a screenshot of a Shiny app*

Description

appshot performs a webshot() using two different methods depending upon the object provided.

If a string is provided (pointing to an 'app.R" file or app directory) an isolated background R process is launched to run the Shiny application. The current R process then captures the webshot().

When a Shiny application object is supplied to appshot, it is reversed: the Shiny application runs in the current R process and an isolated background R process is launched to capture a webshot(). The reason it is reversed in the second case has to do with scoping: although it would be preferable to run the Shiny application in a background process and call webshot from the current process, with Shiny application objects, there are potential scoping errors when run this way.

Usage

```
appshot(
  app,
  file = "webshot.png",
  ...,
  port = getOption("shiny.port"),
  envvars = NULL
)

## S3 method for class 'character'
appshot(
  app,
  file = "webshot.png",
  ...,
  port = getOption("shiny.port"),
  envvars = NULL
)

## S3 method for class 'shiny.appobj'
appshot(
  app,
  file = "webshot.png",
  ...,
  port = getOption("shiny.port"),
  envvars = NULL,
  webshot_timeout = 60
)
```

Arguments

app	A Shiny app object, or a string naming an app directory.
file	A vector of names of output files. Should end with an image file type (.png, .jpg, .jpeg, or .webp) or .pdf. If several screenshots have to be taken and only one filename is provided, then the function appends the index number of the screenshot to the file name. For PDF output, it is just like printing the page to PDF in a browser; selector, cliprect, expand, and zoom will not be used for PDFs.
...	Other arguments to pass on to webshot().
port	Port that Shiny will listen on.
envvars	A named character vector or named list of environment variables and values to set for the Shiny app's R process. These will be unset after the process exits. This can be used to pass configuration information to a Shiny app.
webshot_timeout	The maximum number of seconds the phantom application is allowed to run before killing the process. If a delay argument is supplied (in ...), the delay value is added to the timeout value.

Value

Invisibly returns the normalized path to all screenshots taken. The character vector will have a class of "webshot".

Examples

```
if (interactive()) {
  appdir <- system.file("examples", "01_hello", package="shiny")

  # With a Shiny directory
  appshot(appdir, "01_hello.png")

  # With a Shiny App object
  shinyapp <- shiny::shinyAppDir(appdir)
  appshot(shinyapp, "01_hello_app.png")
}
```

---

resize *Resize an image*

Description

This does not change size of the image in pixels, nor does it affect appearance – it is lossless compression. This requires GraphicsMagick (recommended) or ImageMagick to be installed.

Usage

```
resize(filename, geometry)
```

Arguments

filename	Character vector containing the path of images to resize.
geometry	Scaling specification. Can be a percent, as in "50\%", or pixel dimensions like "120x120", "120x", or "x120". Any valid ImageMagick geometry specification can be used. If filename contains multiple images, this can be a vector to specify distinct sizes for each image.

Value

The filename supplied but with a class value of "webshot".

Examples

```
if (interactive()) {
  # Can be chained with webshot() or appshot()
  webshot("https://www.r-project.org/", "r-small-1.png") %>%
    resize("75%")

  # Generate image that is 400 pixels wide
  webshot("https://www.r-project.org/", "r-small-2.png") %>%
    resize("400x")
}
```

---

rmdshot *Take a snapshot of an R Markdown document*

Description

This function can handle both static Rmd documents and Rmd documents with runtime: shiny.

Usage

```
rmdshot(
  doc,
  file = "webshot.png",
  ...,
  delay = NULL,
  rmd_args = list(),
  port = getOption("shiny.port"),
  envvars = NULL
)
```

Arguments

doc	The path to a Rmd document.
file	A vector of names of output files. Should end with an image file type (.png, .jpg, .jpeg, or .webp) or .pdf. If several screenshots have to be taken and only one filename is provided, then the function appends the index number of the screenshot to the file name. For PDF output, it is just like printing the page to PDF in a browser; selector, cliprect, expand, and zoom will not be used for PDFs.
...	Other arguments to pass on to webshot().
delay	Time to wait before taking screenshot, in seconds. Sometimes a longer delay is needed for all assets to display properly. If NULL (the default), then it will use 0.2 seconds for static Rmd documents, and 3 seconds for Rmd documents with runtime:shiny.
rmd_args	A list of additional arguments to pass to either rmarkdown::render() (for static Rmd documents) or rmarkdown::run() (for Rmd documents with runtime:shiny).
port	Port that Shiny will listen on.
envvars	A named character vector or named list of environment variables and values to set for the Shiny app's R process. These will be unset after the process exits. This can be used to pass configuration information to a Shiny app.

Value

Invisibly returns the normalized path to all screenshots taken. The character vector will have a class of "webshot".

Examples

```
if (interactive()) {
  # R Markdown file
  input_file <- system.file("examples/knitr-minimal.Rmd", package = "knitr")
  rmdshot(input_file, "minimal_rmd.png")

  # Shiny R Markdown file
  input_file <- system.file("examples/shiny.Rmd", package = "webshot")
  rmdshot(input_file, "shiny_rmd.png", delay = 5)
}
```

---

shrink *Shrink file size of a PNG*

Description

This does not change size of the image in pixels, nor does it affect appearance – it is lossless compression. This requires the program optipng to be installed.

Usage

```
shrink(filename)
```

Arguments

filename	Character vector containing the path of images to resize. Must be PNG files.

Details

If other operations like resizing are performed, shrinking should occur as the last step. Otherwise, if the resizing happens after file shrinking, it will be as if the shrinking didn't happen at all.

Value

The filename supplied but with a class value of "webshot".

Examples

```
if (interactive()) {
  webshot("https://www.r-project.org/", "r-shrink.png") %>%
    shrink()
}
```

---

webshot *Take a screenshot of a URL*

Description

Take a screenshot of a URL

Usage

```
webshot(
  url = NULL,
  file = "webshot.png",
  vwidth = 992,
  vheight = 744,
  selector = NULL,
  cliprect = NULL,
  expand = NULL,
  delay = 0.2,
  zoom = 1,
  useragent = NULL,
  max_concurrent = getOption("webshot.concurrent", default = 6),
  quiet = getOption("webshot.quiet", default = FALSE)
)
```

Arguments

`url`	A vector of URLs to visit. If multiple URLs are provided, it will load and take screenshots of those web pages in parallel.
`file`	A vector of names of output files. Should end with an image file type (`.png`, `.jpg`, `.jpeg`, or `.webp`) or `.pdf`. If several screenshots have to be taken and only one filename is provided, then the function appends the index number of the screenshot to the file name. For PDF output, it is just like printing the page to PDF in a browser; `selector`, `cliprect`, `expand`, and `zoom` will not be used for PDFs.
`vwidth`, `vheight`	Viewport width and height. This is the width or height of the virtual browser "window". Chrome expects integer values; numeric values are rounded to the nearest integer.
`selector`	One or more CSS selectors specifying a DOM element to set the clipping rectangle to. The screenshot will contain these DOM elements. For a given selector, if it has more than one match, all matching elements will be used. This option is not compatible with `cliprect`. When taking screenshots of multiple URLs, this parameter can also be a list with same length as `url` with each element of the list containing a vector of CSS selectors to use for the corresponding URL.
`cliprect`	Clipping rectangle. If `cliprect` and `selector` are both unspecified, the clipping rectangle will contain the entire page. This can be the string `"viewport"`, in which case the clipping rectangle matches the viewport size, or it can be a four-element numeric vector specifying the left, top, width, and height. (Note that the order of left and top is reversed from the original webshot package.) This option is not compatible with `selector`. When taking screenshots of multiple URLs, this parameter can also be a list with same length as `url` with each element of the list being "viewport" or a four-elements numeric vector.
`expand`	A numeric vector specifying how many pixels to expand beyond the clipping rectangle determined by `selector`. If one number, the rectangle will be expanded by that many pixels on all sides. If four numbers, they specify the top, right, bottom, and left, in that order. This argument is only applied when `selector` is used and is not compatible with `cliprect`. When taking screenshots of multiple URLs, this parameter can also be a list with same length as `url` with each element of the list containing a single number or four numbers to use for the corresponding URL.
`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% in a desktop browser and 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).

useragent	The User-Agent header used to request the URL.
max_concurrent	(Currently not implemented)
quiet	If TRUE, status updates via console messages are suppressed.

Value

Invisibly returns the normalized path to all screenshots taken. The character vector will have a class of "webshot".

Examples

```
if (interactive()) {

# Whole web page
webshot("https://github.com/rstudio/shiny")

# Might need a delay for all assets to display
webshot("http://rstudio.github.io/leaflet", delay = 0.5)

# One can also take screenshots of several URLs with only one command.
# This is more efficient than calling 'webshot' multiple times.
webshot(c("https://github.com/rstudio/shiny",
          "http://rstudio.github.io/leaflet"),
        delay = 0.5)

# Clip to the viewport
webshot("http://rstudio.github.io/leaflet", "leaflet-viewport.png",
        cliprect = "viewport")

# Specific size
webshot("https://www.r-project.org", vwidth = 1600, vheight = 900,
        cliprect = "viewport")

# Manual clipping rectangle
webshot("http://rstudio.github.io/leaflet", "leaflet-clip.png",
        cliprect = c(200, 5, 400, 300))

# Using CSS selectors to pick out regions
webshot("http://rstudio.github.io/leaflet", "leaflet-menu.png", selector = ".list-group")
# With multiple selectors, the screenshot will contain all selected elements
webshot("http://reddit.com/", "reddit-top.png",
        selector = c("[aria-label='Home']", "input[type='search']"))

# Expand selection region
webshot("http://rstudio.github.io/leaflet", "leaflet-boxes.png",
        selector = "#installation", expand = c(10, 50, 0, 50))

# If multiple matches for a given selector, it will take a screenshot that
# contains all matching elements.
webshot("http://rstudio.github.io/leaflet", "leaflet-p.png", selector = "p")
webshot("https://github.com/rstudio/shiny/", "shiny-stats.png",
         selector = "ul.numbers-summary")
```

```
# Result can be piped to other commands like resize() and shrink()
webshot("https://www.r-project.org/", "r-small.png") %>%
 resize("75%") %>%
 shrink()

}
```

Index