

Package: hexsession (via r-universe)

September 26, 2026

Title Create a Tile of Logos for Loaded Packages

Version 0.1.0

Description Creates a responsive HTML file with tiled hexagonal logos for packages in an R session. Tiles can be also be generated for a custom set of packages specified with a character vector. Output can be saved as a static screenshot in PNG format using a headless browser.

License MIT + file LICENSE

Suggests knitr, rmarkdown, rsvg, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL <https://github.com/luisDVA/hexsession>,
<https://hexsession.liomys.mx/>

BugReports <https://github.com/luisDVA/hexsession/issues>

Depends R (>= 4.1.0)

Imports base64enc, chromote, jsonlite, magick, purrr, htmltools,

Config/pak/sysreqs chromium make libmagick++-dev gsfonts libssl-dev

Repository <https://aleksanderbl29.r-universe.dev>

Date/Publication 2026-04-01 18:39:18 UTC

RemoteUrl <https://github.com/luisDVA/hexsession>

RemoteRef HEAD

RemoteSha 6cbb82163fa27cec063094c3533739f6cf8a6591

Contents

col_arrange	2
encode_image	3
find_imgpaths	3
find_logopaths	4
generate_hexsession_js	4
get_pkg_data	5
getLoaded	6
maincolorRGB	6
make_missingLogos	7
make_tile	7
pkgurls	8
snap_tile	9
Index	10

col_arrange	<i>Arrange Images by Color</i>
-------------	--------------------------------

Description

Takes a vector of image paths, extracts the main color from each image using k-means clustering, converts the colors to the LAB color space, and sorts the images based on the lightness (L) component of their dominant color.

Usage

```
col_arrange(image_paths)
```

Arguments

image_paths Character vector. A vector of file paths to the images.

Value

A character vector of image paths, sorted by the lightness of their main color.

Examples

```
img1 <- system.file("extdata/rectLight.png", package = "hexsession")
img2 <- system.file("extdata/rectMed.png", package = "hexsession")
img3 <- system.file("extdata/rectDark.png", package = "hexsession")
sorted_paths <- col_arrange(c(img1,img3,img2))
```

encode_image	<i>Encode image to Base64</i>
--------------	-------------------------------

Description

Encode image to Base64

Usage

```
encode_image(file_path)
```

Arguments

file_path	Path to an image file
-----------	-----------------------

Value

A character string containing a base64-encoded data URI (e.g., "data:image/png;base64,..."), or NULL with a warning if encoding fails.

Examples

```
img <- system.file("extdata/rectLight.png", package = "hexsession")
encode_image(img)
```

find_imgpaths	<i>Find image paths</i>
---------------	-------------------------

Description

Find image paths

Usage

```
find_imgpaths(pkgnames)
```

Arguments

pkgnames	Character vector of package names
----------	-----------------------------------

Details

Images in svg format will be converted to png. When no image matches 'logo' in the file name the used is will be prompted to select likely logos.

Value

A list of image file paths for each package

find_logopaths	<i>Find logo paths</i>
----------------	------------------------

Description

Find logo paths

Usage

```
find_logopaths(imagepaths, pkgnames)
```

Arguments

imagepaths	List of image paths
pkgnames	Character vector of package names

Value

A vector of logo paths

generate_hexsession_js	<i>Generate JavaScript file for hexsession</i>
------------------------	--

Description

Generate JavaScript file for hexsession

Usage

```
generate_hexsession_js(  
  logopaths,  
  urls,  
  dark_mode,  
  output_js,  
  highlight_mode = FALSE,  
  pkg_names = NULL,  
  focus = NULL  
)
```

Arguments

logopaths	Vector of image paths
urls	Vector of URLs
dark_mode	Use dark mode, inherited from make_tile
output_js	Path to save the JavaScript file
highlight_mode	Use highlight mode, inherited from make_tile
pkg_names	Vector of package names (optional)
focus	Vector of package names to focus on (optional)

Value

Called for its side effect; invisibly returns NULL. Writes a JavaScript file to output_js containing base64-encoded image data and dark/light mode CSS variable assignments.

Examples

```
img <- system.file("extdata/rectLight.png", package = "hexsession")
out <- tempfile(fileext = ".js")
generate_hexsession_js(
  logopaths = img,
  urls = "https://example.com",
  dark_mode = FALSE,
  output_js = out
)
```

get_pkg_data	<i>Get package data</i>
--------------	-------------------------

Description

Get package data

Usage

```
get_pkg_data(packages = NULL)
```

Arguments

packages	Character vector of package names (default is NULL, uses loaded packages)
----------	---

Value

A list containing logopaths, urls, and package names

getLoaded	<i>Get loaded packages</i>
-----------	----------------------------

Description

Get loaded packages

Usage

```
getLoaded()
```

Value

A character vector of the attached packages (excludes base packages)

maincolorRGB	<i>Extract the Most Frequent Color from an Image</i>
--------------	--

Description

Internal helper. For a given image path, this functions uses k-means clustering to identify the most dominant color in the image.

Usage

```
maincolorRGB(imgpath)
```

Arguments

imgpath	Character string. File path to the image.
---------	---

Value

A data frame with one row containing the RGB values of the dominant color. The column name is set to the input image path.

make_missingLogos	Create missing logos
-------------------	----------------------

Description

Create missing logos

Usage

```
make_missingLogos(attached_pkgs, logopath)
```

Arguments

attached_pkgs	Character vector of attached package names
logopath	Vector of existing logo paths

Value

Vector of paths to new logos

make_tile	Generate tile of package logos
-----------	--------------------------------

Description

Creates and returns an interactive html tile of the packages either listed in the packages argument, or all the packages attached to the search path. When rendered interactively, the result is output in the viewer. When rendered in Quarto or RMarkdown, the tile becomes part of the rendered html. revealjs presentations are now supported. If local images are provided, only these images will be used, excluding loaded packages.

Usage

```
make_tile(  
  packages = NULL,  
  local_images = NULL,  
  local_urls = NULL,  
  dark_mode = FALSE,  
  color_arrange = FALSE,  
  highlight_mode = FALSE,  
  focus = NULL,  
  output_dir = tempdir()  
)
```

Arguments

<code>packages</code>	A character vector of package names to include (defaults to NULL, which uses loaded packages)
<code>local_images</code>	Optional character vector of local image paths to add to the tile
<code>local_urls</code>	Optional character vector of URLs for each of the local images passed
<code>dark_mode</code>	Draw the tile on a dark background?
<code>color_arrange</code>	Logical, whether to arrange the images by color along the 'Lab' color space (defaults to FALSE)
<code>highlight_mode</code>	Logical, dim all images except on hover (defaults to FALSE)
<code>focus</code>	A character vector of package names to highlight, dimming all others (defaults to NULL)
<code>output_dir</code>	Directory where the intermediate js, rds, and HTML outputs are written. Defaults to <code>tempdir()</code> . To keep the output HTML in your project folder, pass <code>output_dir = getwd()</code> or any other path. The returned value and a console message both show the exact file location.

Details

If an installed package does not have a bundled logo, or if the logo has been added to `.Rbuildignore`, a generic logo with the name of the package will be created instead. When the function cannot locate a package logo unambiguously, users will be prompted to select one from a list of potential options.

Set the execution options to `output: asis` in Quarto revealjs presentations to enable raw markdown output and adequate rendering of the tiles.

Value

The path to the output HTML file (invisibly) when called interactively. A `message()` also prints the location. Returns an `htmltools` HTML object when rendered inside Quarto or R Markdown.

Examples

```
img1 <- system.file("extdata/rectLight.png", package = "hexsession")
img2 <- system.file("extdata/rectMed.png", package = "hexsession")
img3 <- system.file("extdata/rectDark.png", package = "hexsession")
path <- make_tile(local_images = c(img1, img2, img3))
```

pkgurls

Get package URLs

Description

Get package URLs

Usage

```
pkgurls(pkgnames)
```

Arguments

pkgnames Character vector of package names

Value

A vector of package URLs

snap_tile	<i>Take screenshot of html image tile</i>
-----------	---

Description

Take screenshot of html image tile

Usage

```
snap_tile(
  output_path,
  screen_width = 800,
  screen_height = 700,
  dark_mode = FALSE,
  output_dir = getwd()
)
```

Arguments

output_path Path to image file

screen_width Width of the browser window

screen_height Height of the browser window

dark_mode Is the tile being saved dark or light mode?

output_dir Directory where make_tile() wrote its files. Must match the output_dir used in the preceding make_tile() call. Defaults to getwd(). To snap a tile written to tempdir(), pass output_dir = tempdir() explicitly.

Value

Path to the saved PNG image (the value of output_path).

Examples

```
snap_tile(tempfile(fileext = ".png"))
```

Index

`col_arrange`, [2](#)

`encode_image`, [3](#)

`find_imgpaths`, [3](#)

`find_logopaths`, [4](#)

`generate_hexsession_js`, [4](#)

`get_pkg_data`, [5](#)

`getLoaded`, [6](#)

`maincolorRGB`, [6](#)

`make_missingLogos`, [7](#)

`make_tile`, [7](#)

`pkgurls`, [8](#)

`snap_tile`, [9](#)