Package: hexsession (via r-universe)

February 13, 2025

Title Create a tile of logos for loaded packages

Version 0.0.0.9000

Description Creates a responsive HTML file with tiled hex logos for all loaded packages in a session, which can be saved as a static screenshot in png format.

License MIT + file LICENSE

Suggests rsvg, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2.9000

URL https://github.com/luisDVA/hexsession, https://luisdva.github.io/hexsession/

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

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

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

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

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

RemoteRef HEAD

RemoteSha e693222adf9ab4fc749fd98729ebe6c369556d0c

Contents

`col_arrange` *Arrange Images by Color*

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 *Encode image to Base64*

Description

Encode image to Base64

Usage

```
encode_image(file_path)
```

Arguments

file_path	Path to an image file

Value

Base64 encoded string of the image

find_imgpaths *Find image paths*

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 *Find logo paths*

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 *Generate JavaScript file for hexsession*

Description

Generate JavaScript file for hexsession

Usage

```
generate_hexsession_js(logopaths, urls, dark_mode, output_js)
```

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

getLoaded *Get loaded packages*

Description

Get loaded packages

Usage

```
getLoaded()
```

Value

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

get_pkg_data *Get package data*

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 and urls for the packages

maincolorRGB *Extract the Most Frequent Color from an Image*

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, logopaths)
```

Arguments

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

Value

Vector of paths to new logos

make_tile *Generate tile of package logos*

Description

This function returns an interactive html tile view of the packages either listed in the packages option, or all of the loaded packages. When rendered interactively, the result is output in the viewer. When rendered in Quarto or RMarkdown, the tile becomes part of the rendered html. 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
)
```

Arguments

packages	Character vector of package names to include (default: NULL, which uses loaded packages)
local_images	Optional character vector of local image paths to add to the tile
local_urls	Optional character vector of URLs for each of the local images passed
dark_mode	Draw the tile on a dark background?
color_arrange	Logical, whether to arrange the images by color along the 'Lab' color space (defaults to FALSE)

Value

Path to the output file

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 *Take screenshot of html image tile*

Description

Take screenshot of html image tile

Usage

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

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?

Value

Path to the saved image

Index