Package: Rigma (via r-universe)

November 5, 2024

Type Package

Title Access to the 'Figma' API

Version 0.3.9000

Maintainer Alexandros Kouretsis <alexandros@appsilon.com>

Description The goal of Rigma is to provide a user friendly client to the 'Figma' API <https://www.figma.com/developers/api>. It uses the latest `httr2` for a stable interface with the REST API. More than 20 methods are provided to interact with 'Figma' files, and teams. Get design data into R by reading published components and styles, converting and downloading images, getting access to the full 'Figma' file as a hierarchical data structure, and much more. Enhance your creativity and streamline the application development by automating the extraction, transformation, and loading of design data to your applications and 'HTML' documents.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://github.com/AleKoure/Rigma

RoxygenNote 7.2.1

Imports bslib, checkmate, dplyr, fs, glue, httr2, lubridate, magrittr, png, purrr, rlang, tibble, tidyr, withr, xml2

Suggests covr, httptest2, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs make libicu-dev libpng-dev libxml2-dev libssl-dev

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

RemoteUrl https://github.com/alekoure/rigma

RemoteRef HEAD

RemoteSha eeab4e378feb0d8f0a011ba4e08bfe1974f4c598

Contents

`add_color` *Add color to design table data*

Description

Extracts color for design data collected from the Figma API. Can convert to hex. Infered color data are added in a new column named as ‘color‘.

Usage

```
add_color(design_tibble, hex = TRUE)
```

Arguments

design_tibble	Tabular data to be augmented with color colomn.
hex	logical. If 'TRUE' hex data are added to 'color' column else nested data for 'RGBA' channels.

Value

Adds color column to 'design_tibble' data

Examples

```
## Not run:
file_key <- "sFHgQh9dL6369o5wrZHmdR"
resp <- get_file_styles(file_key) %>%
  as_design_tibble() %>%
  add_color()

## End(Not run)
```

---

as_design_tibble *Transform data to tabular format*

Description

Transforms data returned from a Figma API request to tabular format.

Usage

```
as_design_tibble(rigma_resp, message = TRUE)
```

Arguments

rigma_resp	A response object from Figma API.
message	logical. Control printing of messages.

Value

tibble data extracted from Figma response objects. Subclasses of type 'design_tibble' are added to the resulting tibbles.

Examples

```
## Not run:
file_key <- "sFHgQh9dL6369o5wrZHmdR"
resp <- get_file_styles(file_key = file_key)
as_design_tibble(resp)

## End(Not run)
```

delete_comment *DELETE comments*

Description

Deletes a specific comment. Only the person who made the comment is allowed to delete it.

Usage

```
delete_comment(file_key, comment_id)
```

Arguments

file_key	string. The key that a Figma file is referred by.
comment_id	string. The comment id.

Value

S3 object of class 'rigma_delete_comment'. Contains the parsed JSON response with fields 'error', 'status', and 'i18n'.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
first_comment_id <- get_comments(file_key)$comments[[1]]$id
delete_comment(file_key, first_comment_id)

## End(Not run)
```

delete_comment_reactions *DELETE comment reactions*

Description

Removes a particular comment reaction. The only person with the ability to delete a comment reaction is the original poster.

Usage

```
delete_comment_reactions(file_key, comment_id, emoji)
```

Arguments

file_key	string. The key that a Figma file is referred by.
comment_id	string. Comment id of comment to delete reaction from.
emoji	string. The emoji type of reaction to delete as a string enum :eyes:, :heart_eyes:, :heavy_plus_sign:, :+1:, :-1:, :joy: and :fire:

Value

S3 object of class ‘rigma_delete_comment_reactions‘. Contains the parsed JSON response with fields ‘error‘, ‘status‘, and ‘i18n‘.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
first_comment_id <- get_comments(file_key)$comments[[1]]$id
delete_comment_reactions(file_key, first_comment_id, ":eyes:")

## End(Not run)
```

---

extract_bslib_palette *Extract bslib palette*

Description

Extract bslib palette

Usage

```
extract_bslib_palette(design_tibble, version = 5)
```

Arguments

design_tibble	A design_tibble returned bu the ‘add_color()‘ function
version	Bootstrap version to use for extracting color variables

Details

This function filters for bslib high level color variables published as styles of a Figma file and/or team. The color should be extracted and added as a variable to the retrieved data.

Value

List with colors used in high level variables of ‘bs_theme()‘.

Examples

```
## Not run:
file_key <- "sFHgQh9dL6369o5wrZHmdR"
file_key %>%
  get_file_styles() %>%
  as_design_tibble() %>%
  add_color() %>%
  extract_bslib_palette()

## End(Not run)
```

`get_comments` *GET comments*

Description

Gets a list of comments left on the file.

Usage

```
get_comments(file_key)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.

Value

S3 object of class ‘rigma_get_comments‘. Contains the parsed JSON response.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_comments(file_key)

## End(Not run)
```

get_comments_reactions *GET comments reactions*

Description

Obtains a paginated list of the comments' reactions.

Usage

```
get_comments_reactions(file_key, comment_id, cursor = NULL)
```

Arguments

file_key	string. The key that a Figma file is referred by.
comment_id	string. The comment id.
cursor	string obtained from the result of the previous request, a cursor for pagination.

Value

S3 object of class 'rigma_get_comments_reactions'. Contains the parsed JSON response with fields 'reactions', and 'pagination'.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
resp <- get_comments(file_key)
resp_reactions <- get_comments_reactions(file_key, resp$comments[[1]]$id)

## End(Not run)
```

get_component *GET component*

Description

Get metadata on a component by key. To publish components join a Figma team and subscribe for professional account.

Usage

```
get_component(key)
```

Arguments

key	string. The unique identifier of the component.

Value

S3 object of class ‘rigma_get_component‘.

Examples

```
## Not run:
component_key <- "my_key"
get_component(component_key)

## End(Not run)
```

---

get_component_sets *GET component set*

Description

Get metadata on a component_set by key. To publish components join a Figma team and subscribe for professional account.

Usage

```
get_component_sets(key)
```

Arguments

key	string. The unique identifier of the component.

Value

S3 object of class ‘rigma_get_component_set‘.

Examples

```
## Not run:
component_key <- "my_key"
get_component_sets(component_key)

## End(Not run)
```

get_file *GET file*

Description

Direct access to the desired file is made available using the GET file API endpoint. It returns a JSON object representing the file pointed to by:key. Any Figma file url, such as https://www.figma.com/file/:key/:title, can be used to parse the file key.

Usage

```
get_file(
  file_key,
  version = NULL,
  ids = NULL,
  depth = NULL,
  geometry = NULL,
  plugin_data = NULL,
  branch_data = NULL
)
```

Arguments

file_key	string. The key that a Figma file is referred by.
version	string. A certain version ID to obtain. By omitting this, you'll obtain the file's most recent version.
ids	string. list the document's nodes that are important to you, separated by commas. If supplied, only the nodes listed, their children, and everything between the root node and the listed nodes will be returned as part of the document.
depth	integer. A positive number indicating the depth of the traversal across the document tree. For instance, changing this to 2 returns both Pages and all top level objects on each page instead of just returning Pages. All nodes are returned if this argument is not set.
geometry	string. To export vector data, set equal to "paths".
plugin_data	string. A list of plugin IDs separated by commas or the word "shared." The result's 'pluginData' and 'sharedPluginData' attributes will contain any data existing in the document created by those plugins.
branch_data	boolean. The requested file's branch metadata is returned. If the file is a branch, the returned response will also provide the key for the main file. If the file has branches, the response will also contain the metadata for those branches. Standard: false.

Value

S3 object of class ‘figma_file_resp‘. Contains the parsed content, the path, and the API response compatible with ‘httr2‘ methods. The retrieved file’s metadata includes the ‘name‘, ‘lastModified‘, ‘thumbnailUrl‘, ‘editorType‘, ‘linkAccess‘, and ‘version attributes‘. A Node with the DOCUMENT type is present in the document attribute.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_file(file_key)

## End(Not run)
```

`get_file_components` *GET file components*

Description

Get a list of published components within a file library. Note that published components are only available via the professional plan.

Usage

```
get_file_components(file_key)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.

Value

S3 object of class ‘rigma_get_file_components‘. Components are stored in the ‘meta‘ field.

Examples

```
## Not run:
#navigate to team page and get id from url
file_key <- "gYRjH0y8ZM0VtEfO8kf6ch"
get_file_components(file_key)

## End(Not run)
```

get_file_component_sets *GET file component sets*

Description

Get a list of published component_sets within a file library. Note that published components are only available via the professional plan.

Usage

```
get_file_component_sets(file_key)
```

Arguments

file_key	string. Id of the team to list components from.

Value

S3 object of class 'rigma_get_file_component_sets'. Components are stored in the 'meta' field.

Examples

```
## Not run:
#navigate to team page and get id from url
file_key <- "gYRjH0y8ZM0VtEfO8kf6ch"
get_file_component_sets(file_key)

## End(Not run)
```

get_file_nodes *GET file nodes*

Description

Returns a JSON object containing the nodes referenced by ':ids'. The Figma file referred to by ':key' is where the nodes are located. The node Id and file key can be parsed from any Figma node url: https://www.figma.com/file/:key/:title?node-id=:id.

Usage

```
get_file_nodes(
  file_key,
  ids = NULL,
  version = NULL,
  depth = NULL,
  geometry = NULL,
  plugin_data = NULL
)
```

Arguments

file_key	string. The key that a Figma file is referred by.
ids	character. Vector with the document's node ids that are important to you. If supplied, only the nodes listed, their children, and everything between the root node and the listed nodes will be returned as part of the document.
version	string. A certain version ID to obtain. By omitting this, you'll obtain the file's most recent version.
depth	integer. A positive number indicating the depth of the traversal across the document tree. For instance, changing this to 2 returns both Pages and all top level objects on each page instead of just returning Pages. All nodes are returned if this argument is not set.
geometry	string. To export vector data, set equal to "paths".
plugin_data	string. A list of plugin IDs separated by commas or the word "shared." The result's 'pluginData' and 'sharedPluginData' attributes will contain any data existing in the document created by those plugins.

Value

S3 object of the type 'rigma_get_file_nodes'.

The supplied file's metadata includes the 'name', 'lastModified', 'thumbnailUrl', 'editorType', and 'version' attributes.

The file link share permission level is described in the 'linkAccess' field. A shared link may have one of five different permissions: "inherit," "view," "edit," "org view," and "org edit." The default permission for files produced in a team project is "inherit," and those files will take on the project's rights by default. "org view" and "org edit" only allow org users to access the link.

Each node has the ability to inherit properties from applicable styles. A mapping from style IDs to style metadata is contained in the 'styles' key.

It's important to note that the nodes field is a list that can include null values. This can be because the provided file does not contain a node with the specified id.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
```

```
get_file_nodes(file_key, ids = "0:0")

## End(Not run)
```

`get_file_styles` *GET file styles*

Description

Get published styles in a file library by name.

Usage

```
get_file_styles(file_key)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.

Value

S3 object of class ‘rigma_get_file_styles‘. Styles are stored in the ‘meta‘ field.

Examples

```
## Not run:
#navigate to team page and get id from url
file_key <- "gYRjH0y8ZM0VtEfO8kf6ch"
get_file_styles(file_key)

## End(Not run)
```

`get_file_versions` *GET file versions*

Description

A list of the versions of a file.

Usage

```
get_file_versions(file_key)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.

Value

S3 object of class ‘rigma_get_file_versions‘. Contains the parsed JSON response with fields ‘versions‘, and ‘pagination‘.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_file_versions(file_key)

## End(Not run)
```

`get_image` *GET image*

Description

If there are no errors, "images" will be filled with a list of node IDs and their corresponding URLs for the displayed images, and "status" will be left empty. After 30 days, the picture assets will stop working.

It's important to note that the picture map could include null values. This suggests that the node's attempt to render has failed. The lack of a node id or other factors, such as the absence of renderable components, may be responsible. Whether or not the render was successful, it is assured that any node that was requested for rendering will be represented in this map.

Usage

```
get_image(
  file_key,
  ids,
  scale = NULL,
  format = NULL,
  svg_include_id = NULL,
  svg_simplify_stroke = NULL,
  use_absolute_bounds = NULL,
  version = NULL
)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.
`ids`	string. A comma separated list of node IDs to render
`scale`	numeric. The image scaling factor is a number between 0.01 and 4

format string. A string enum for the image output format, can be "jpg", "png", "svg", or "pdf"

svg_include_id logical. Whether or not to give each SVG element an id attribute. Standard: 'FALSE'

svg_simplify_stroke
logical. Whether to simplify inside/outside strokes and use stroke attribute if possible instead of <mask>. Default: 'TRUE'.

use_absolute_bounds
logical. Whether the node is cropped or the area around it is vacant, use the node's full dimensions. To export text nodes without cropping, use this method. Standard: false.

version string. A specific version ID to use. Omitting this will use the current version of the file

Value

S3 object of class 'rigma_get_image'. Contains the parsed JSON response with fields 'err', and 'images'.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_image(file_key, ids = "0:1", format = "svg")

## End(Not run)
```

---

get_image_fills *GET image fills*

Description

This API endpoint provides download URLs for all of the images that are present in a document's image fills. Figma represents any user-supplied images using image fills. When you drag a picture into Figma, it builds a rectangle with a single fill to represent it. The user can then modify the attributes of the fill and change the rectangle as they see fit.

Usage

```
get_image_fills(file_key)
```

Arguments

file_key string. The key that a Figma file is referred by.

Value

This API provides a mapping from image references to download URLs for the images. Image URLs have a 14-day maximum lifespan. Image references are located in the output of the GET files endpoint under the ‘imageRef‘ attribute in a Paint.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_image_fills(file_key)

## End(Not run)
```

---

get_project_files *GET project files*

Description

List the files in a given project.

Usage

```
get_project_files(project_id, branch_data = "false")
```

Arguments

project_id	string. Id of the project to list files from.
branch_data	string. Returns branch metadata in the response for each main file with a branch inside the project. Default: "false"

Value

S3 object of class ‘rigma_get_project_files‘. Contains the parsed JSON response with fields ‘name‘, and ‘files‘.

Examples

```
## Not run:
#navigate to  file and get key from url
project_id <- "71686204"
get_project_files(project_id)

## End(Not run)
```

get_team_components *GET team components*

Description

Get a paginated list of published components within a team library. Note that team components are only available via the professional plan.

Usage

```
get_team_components(team_id, page_size = 30, after = NULL, before = NULL)
```

Arguments

team_id	string. The team's ID, with a list of its components.
page_size	number Number of items in a paged list of results. Defaults to 30.
after	number. Cursor providing the id for which to begin component retrieval. Exclusive with 'before' parameter. The cursor value is a tracked integer that is kept internally but has no Ids.
before	number. The id before which to begin obtaining components is shown by the cursor. Exclusive with after The cursor value is a tracked integer that is kept internally but has no Ids.

Value

S3 object of class 'rigma_get_team_components'. Components are stored in the 'meta' field.

Examples

```
## Not run:
#navigate to team page and get id from url
get_team_components(team_id = "1168610438838663284")

## End(Not run)
```

get_team_component_sets *GET team component sets*

Description

Get a paginated list of published 'component_sets' within a team library. Note that published components are only available via the professional plan.

Usage

```
get_team_component_sets(team_id, page_size = 30, after = NULL, before = NULL)
```

Arguments

team_id	string. The team's ID, with a list of its components.
page_size	number Number of items in a paged list of results. Defaults to 30.
after	number. Cursor providing the id for which to begin component retrieval. Exclusive with 'before' parameter. The cursor value is a tracked integer that is kept internally but has no Ids.
before	number. The id before which to begin obtaining components is shown by the cursor. Exclusive with after The cursor value is a tracked integer that is kept internally but has no Ids.

Value

S3 object of class 'rigma_get_team_component_sets'. Components are stored in the 'meta' field.

Examples

```
## Not run:
#navigate to team page and get id from url
get_team_component_sets(team_id = "1168610438838663284")

## End(Not run)
```

---

get_team_projects *GET team projects*

Description

Get a list of all the Projects inside the specified team using this Endpoint. Only projects visible to the authorized user or the holder of the developer token will be returned. It should be noted that a user's team ID cannot yet be determined from a token. Go to the team page of the team you are a part of to get your team ID. After the term "team" and before your team name, the team id will appear in the URL.

Usage

```
get_team_projects(team_id)
```

Arguments

team_id	string. Id of the team to list projects from.

Value

S3 object of class ‘rigma_get_team_projects‘. Contains the parsed JSON response with fields ‘name‘, and ‘projects‘.

Examples

```
## Not run:
#navigate to  file and get key from url
team_id <- "1168610438838663284"
get_team_projects(team_id)

## End(Not run)
```

`get_team_styles` *GET team styles*

Description

Get a list of published styles in a team library that is paginated.

Usage

```
get_team_styles(team_id, page_size = 30, after = NULL, before = NULL)
```

Arguments

`team_id`	string. The team's ID, with a list of its styles.
`page_size`	number. Number of items in a paged list of results. Defaults to 30.
`after`	number. Cursor providing the id for which to begin component retrieval. Exclusive with ‘before‘ parameter. The cursor value is a tracked integer that is kept internally but has no Ids.
`before`	number. The id before which to begin obtaining components is shown by the cursor. Exclusive with after The cursor value is a tracked integer that is kept internally but has no Ids.

Value

S3 object of class ‘rigma_get_team_styles‘. Styles are stored in the ‘meta‘ field.

Examples

```
## Not run:
#navigate to team page and get id from url
get_team_styles(team_id = "1168610438838663284")

## End(Not run)
```

possibly_thumbnail_color *Possibly extract thumbnail color*

Description

Possibly extract thumbnail color

Usage

```
possibly_thumbnail_color(...)
```

Arguments

...	Variables passed to wrapped function

Value

thumbnail_color function that handles errors

post_comment *POST comment*

Description

Posts a new comment on the file.

Usage

```
post_comment(file_key, message, comment_id = NULL, client_meta)
```

Arguments

file_key	string. The key that a Figma file is referred by.
message	string. The comment's textual content to post.
comment_id	string. If there is one, the comment to respond to. You cannot reply to a remark that is a reply to itself (a reply has a parent id), thus this must be a root comment.
client_meta	string. The position of where to place the comment.

Value

S3 object of class 'rigma_post_comment'. Contains the parsed JSON response with fields 'id', 'file_key', 'status', 'i18n', 'parent_id', 'user', 'created_at', 'resolved_at', 'message', 'reactions', 'client_meta', and 'order_id'.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
get_comments(file_key)

## End(Not run)
```

`post_comment_reactions` *POST comment reactions*

Description

Posts a new comment reaction on a file comment.

Usage

```
post_comment_reactions(file_key, comment_id, emoji)
```

Arguments

`file_key`	string. The key that a Figma file is referred by.
`comment_id`	string. If there is one, the comment to respond to. You cannot reply to a remark that is a reply to itself (a reply has a parent id), thus this must be a root comment.
`emoji`	string. The emoji type of reaction as a string enum :eyes:, :heart_eyes:, :heavy_plus_sign:, :+1:, :-1:, :joy: and :fire:

Value

S3 object of class 'rigma_post_comment_reactions'. Contains the parsed JSON response with fields 'error', 'status', and 'i18n'.

Examples

```
## Not run:
#navigate to  file and get key from url
file_key <- "sFHgQh9dL6369o5wrZHmdR"
first_comment_id <- get_comments(file_key)$comments[[1]]$id
post_comment_reactions(file_key, first_comment_id, ":eyes:")

## End(Not run)
```

req_rigma_agent *Add metadata to Rigma request*

Description

Extra metadata attached to httr2 request object before performing the request.

Usage

```
req_rigma_agent(req, user_agent = "Rigma https://github.com/AleKoure/Rigma")
```

Arguments

req	httr2_request object
user_agent	string with user agent

Value

httr2_request object with Figma token heater, user agent and retry specifications.

safe_download *Safe download files*

Description

Safe download files

Usage

```
safe_download(...)
```

Arguments

...	Variables passed to wrapped function

Value

download function that safely handles errors

text_data_from_styles *Get text data from styles*

Description

Get text data from styles

Usage

```
text_data_from_styles(design_tibble)
```

Arguments

design_tibble	S3 object of class 'design_tibble_style' returned by queering the 'Figma' API for the published styles. Its design tibble should contain a unique 'file_key' (map for each 'file_key' if more than one exist.)

Details

Given a design_tibble with exported styles this function retrieves all TEXT style type metadata. It uses the GET file nodes API endpoint and collects all style data.

Value

tibble with text metadata of exported TEXT styles

Examples

```
## Not run:
file_key <- "sFHgQh9dL6369o5wrZHmdR"
file_key %>%
 get_file_styles() %>%
 as_design_tibble() %>%
 text_data_from_styles()

## End(Not run)
```

thumbnail_color *Find thumbnail color*

Description

Find thumbnail color

Usage

```
thumbnail_color(path, hex = TRUE)
```

Arguments

path	string. Path to the thumbnail PNG
hex	logical. If 'TRUE' then the RGBA values are converted to hex

Details

Given a mono-colored thumbnail this function extracts the RGBA channels and returns a vector scaled from [0, 1] or a hex color code.

Value

The color of the thumbnail in hex or rgba

Examples

```
path <- system.file("extdata", "test_thumbnail.png", package = "Rigma")
thumbnail_color(path)
```

Index