

Package: figma (via r-universe)

August 25, 2024

Type Package

Title Web Client/Wrapper to the 'Figma API'

Version 0.2.0

Author Pedro Faria [aut, cre, cph]

Maintainer Pedro Faria <pedropark99@gmail.com>

URL <https://github.com/pedropark99/figma>,
<https://pedropark99.github.io/figma/>

BugReports <https://github.com/pedropark99/figma/issues>

Description An easy-to-use web client/wrapper for the 'Figma API' <<https://www.figma.com/developers/api>>. It allows you to bring all data from a 'Figma' file to your 'R' session. This includes the data of all objects that you have drawn in this file, and their respective canvas/page metadata.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports httr (>= 1.4.1), purrr (>= 0.3.3), dplyr (>= 1.0.0), tidyr (>= 1.0.0), rlang (>= 1.0.0), tibble (>= 3.0.5)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, usethis, emoji, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 4.1)

Config/testthat/edition 3

Repository <https://pedropark99.r-universe.dev>

RemoteUrl <https://github.com/pedropark99/figma>

RemoteRef HEAD

RemoteSha 89b281e0ce8333ad48ee221ec69d2edaf6316848

Contents

as_figma_document	2
as_tibble	3
build_query_string	4
build_request_url	5
default_attrs	6
document_attrs	6
get_document_info	7
get_endpoint_url	8
get_figma_file	9
get_figma_page	11
parse_response_object	13
quarto_website	13
select_any_of	14
untitled_file	15

Index	17
--------------	-----------

as_figma_document	<i>Convert a httr response object to a Figma Document object</i>
-------------------	--

Description

This function receives a `httr::response()` object, and outputs a `figma_document` object.

Usage

```
as_figma_document(response, ...)
```

Arguments

response	a response object produced by a httr HTTP method (e.g. <code>httr::GET()</code> , <code>httr::POST()</code> , etc.);
...	Further arguments passed by the caller. Currently ignored by the function;

Details

A Figma Document is a just a R list with a more organized structure than the raw content of the HTTP request in `httr::response()`. You can access each part of this list with ``$`` and ``[[`` operators. See Value section to understand what is in each element of this list.

`as_figma_document()` will call different parsers depending on what kind of elements are present in the content of the response object it receives as input. These elements define what kind of data is present in the response object, and how it is structured.

If this input object have a document element in the top-level of the content, is likely that this input object was produced by `figma::get_figma_file()`. In this case, `as_figma_document()` will call `figma::parse_figma_file()` to parse the contents of the HTTP request.

In the other hand, if this input object have a nodes element in the top-level of the content, is likely that this input object was produced by `figma::get_figma_page()`. In this case, `as_figma_document()` will call `figma::parse_figma_page()` to parse the contents of the HTTP request.

If none of these key elements ("document" or "nodes") are found in the top-level of the content of the response object, `as_figma_document()` will issue an error to the user, telling it could not recognize the source of the response object.

Value

Returns an object of class `figma_document`, which is a R list with these elements:

- **document**: a R list with all of the document metadata of your Figma file;
- **canvas**: a R list with all of the canvas and objects data of your Figma file;
- **n_canvas**: a integer with the number of canvas/pages in your Figma file;
- **n_objects**: a vector of integers with the number of objects in each canvas/page of your Figma file;

Examples

```
## Not run:
file_key <- "hch8YlkgauIZ9raDzjPvCz"
token <- "my figma token secret ..."
# Returns a `response` object:
r <- figma::get_figma_file(file_key, token)
result <- figma::as_figma_document(r)

## End(Not run)
```

as_tibble

Convert a httr response object to a tibble object

Description

This function receives a `httr::response()` object, and outputs a `tibble::tibble()` object.

Usage

```
as_tibble(x, ...)
```

Arguments

- `x` A Figma document object (i.e. a `figma::figma_document` object), or, a `httr::response` object to parse;
- `...` Further arguments passed by the caller. Only simplified argument is currently accepted, other arguments are ignored (See Details section);

Details

The function parses the data from the response object and tries to fit it into a tibble object. Each row in the resulting tibble will be describing an object in your Figma file.

If `as_tibble()` receives a response object as input, it will call `as_figma_document()` to convert this response object into a `figma_document` object. But, `as_tibble()` can receive directly a `figma_document` object and jump this step.

If the Figma file have no objects draw in a specific canvas, `as_tibble()` will return an empty tibble object for this specific canvas. This means that, if your Figma file is empty, or, in other words, if all of the canvas/pages of the file have no objects draw in them, the final result of `as_tibble()` will be an empty tibble object.

By default, `figma::as_tibble()` does not include any document metadata in the resulting tibble object. But you can pass `simplified = FALSE` to the function to change this behavior.

Value

A `tibble::tibble()` object with all of the canvas and objects data of your Figma file.

Examples

```
## Not run:
file_key <- "hch8Y1kgaUIZ9raDzjPvCz"
token <- "my figma token secret ... "
# Returns a `response` object:
r <- figma::get_figma_file(file_key, token)
result <- figma::as_tibble(r)

# To include all of the document metadata, use `simplified = FALSE`
result <- figma::as_tibble(r, simplified = FALSE)

## End(Not run)
```

<code>build_query_string</code>	<i>Build a query string from a set of named parameters</i>
---------------------------------	--

Description

Utility function used to build query strings (non-exported function).

Usage

```
build_query_string(parameters)
```

Arguments

`parameters` A list with a set of key-value pairs to compose the query string

Details

This function takes a set of key-value pairs (or in other words, a set of named arguments), to build a query string. It basically combine (or "collapse") all key-value pairs together, to form the resulting query string.

Logical values (TRUE or FALSE) are automatically converted to a lower-case version ("true" or "false"), since these versions are more typically used in standard query strings.

Value

A single string with the query string produced.

build_request_url	<i>Build the request URL</i>
-------------------	------------------------------

Description

Add multiple "components" to a base URL, to build the complete URL that will be used in the HTTP request (non-exported function).

Usage

```
build_request_url(base_url, path = NULL, ...)
```

Arguments

base_url	A single string with the base URL that you want add components to;
path	A vector of strings (or a single string) with "path" components;
...	Key-value pairs that will compose the query string section of the URL;

Details

This function receives as input, a set of pieces (or components) of the URL that will be used in the HTTP request. Then, it tries to combine (or "collapse") all these pieces together to form a single string with the complete URL.

There are three main types of pieces (or components) accepted by this function. First, the base URL, which is the initial portion of the URL. Usually, this is the base URL for the Figma API.

Second, we have the "path" components, which are all the small bits that compose the path and resource sections of the URL. Each element of the vector given to path is separated by a slash character ("/") in the final result.

For example, if I give the vector c("path1", "path2", "path3") to path, the end result will be structured like this:

```
base_url/path1/path2/path3
```

Third, a query string, which is usually composed by a set of key-value pairs. build_request_url() collects all these key-value pairs through the ... argument, and then, combines all these pairs together to form a query string.

Value

A single string with the complete URL.

default_attrs	<i>Default attributes of every Figma node</i>
---------------	---

Description

Every Figma document is represented as a three of nodes, and, each node have a different type (e.g. DOCUMENT, CANVAS, TEXT, etc.). However, despite their differences in type, every node always has three default attributes, which are stored in the `figma::default_attrs` object.

Usage

```
default_attrs
```

Format

An object of class character of length 3.

document_attrs	<i>Default document attributes of a Figma file</i>
----------------	--

Description

Every Figma file have some default document attributes, that is, some informations that describe the file, like its name, the last time it was modified, etc. These default attributes are stored in the `figma::document_attrs` object.

Usage

```
document_attrs
```

Format

An object of class character of length 11.

get_document_info	<i>Get the document metadata of a Figma File from the API</i>
-------------------	---

Description

This function uses the `/v1/files/` endpoint of Figma API to get all of the document metadata of a particular Figma file, and fit it into a R object.

Usage

```
get_document_info(file_key, token, .output_format = "list")
```

Arguments

`file_key` A string with the key of the Figma File you want to get;
`token` A string with your personal Figma token to authenticate in the API;
`.output_format` The output format. Options are "list" and "tibble". Defaults to "list";

Details

You may not be interested in the contents of a Figma file, but in the metadata of this file instead. That is, you want to know the "name" of a particular Figma file, the last time it was modified, which version it uses, etc.

That is why `get_document_info()` exists. It collects just the metadata of your Figma file, and ignores all canvas and objects data.

By default, `get_document_info()` fits the metadata into a raw R list. But you can change this behavior with the `.output_format` argument. With `.output_format = "tibble"`, `get_document_info()` will fit the metadata into a `tibble::tibble` object.

Value

By default, `get_document_info()` returns a raw R list with all of the document metadata of your Figma file. But you can change this behavior with `.output_format = "tibble"`, which gives you a `tibble::tibble` object.

Be aware of possible HTTP errors

To get the data of your Figma file, the functions from `figma` package make a HTTP request to the Figma API. But this request can fail for a number of reasons, and if this does happen, `get_document_info()` will use `report_http_error()` to raise an error and report to the user, what kind of error message the Figma API returned. See `vignette("http-errors")` for more details.

Examples

```
## Not run:
library(figma)
file_key <- "hch8YlkgaUIZ9raDzjPvCz"
token <- "my figma token secret ..."
# Returns a list with the metadata:
result <- figma::get_document_info(file_key, token)
# Returns a `tibble` object:
result <- figma::get_document_info(
  file_key, token,
  .output_format = "tibble"
)

## End(Not run)
```

get_endpoint_url	<i>Get the URL to a endpoint of Figma API</i>
------------------	---

Description

Get the URL to a endpoint of Figma API

Usage

```
get_endpoint_url(endpoint = NULL)
```

Arguments

endpoint	A single string with the name of the desired endpoint (needs to be one of the values present in <code>figma::implemented_endpoints</code>). Defaults to <code>NULL</code> ;
----------	--

Details

If the function is called without any arguments, `get_endpoint_url()` will output a list with all of the implemented endpoints.

However, the function accepts a single string value with the name of, a specific endpoint. In this case, `get_endpoint_url()` will output a single string with the endpoint you selected. Is worth mentioning, that this string must be one of the values present in `figma::implemented_endpoints`.

If the user provided any type of value that does not fit in this description, the function will prompt the user with an error message.

Value

A string with the URL to the given endpoint, or, a list with all of the implemented endpoints;

Examples

```
# Returns the URL to the `files` endpoint of Figma API
library(figma)
figma::get_endpoint_url("files")
```

get_figma_file

Get data of a Figma File from the API

Description

This function uses the `/v1/files/` endpoint of Figma API to get all of the data of a particular Figma file, and fit it into a R object.

Usage

```
get_figma_file(
  file_key,
  token,
  geometry = FALSE,
  .output_format = "response",
  ...
)
```

Arguments

<code>file_key</code>	A string with the key of the Figma File you want to get;
<code>token</code>	A string with your personal Figma token to authenticate in the API;
<code>geometry</code>	A boolean value indicating if you want to export vector data. Defaults to <code>FALSE</code> ;
<code>.output_format</code>	The output format. Options are <code>"response"</code> , <code>"figma_document"</code> , <code>"tibble"</code> . Defaults to <code>"response"</code> ;
<code>...</code>	Further arguments that are passed to <code>parse_response_object()</code> ;

Details

With this function you can bring all of the data of your Figma file into R. By default, `get_figma_file()` returns a `'response'` object with all of the data returned by the API. That is, not only the data of your Figma file, but also, the data from the HTTP request.

All of your Figma file data is in the `content` element of the `'response'` object. However, by default, the Figma API returns this data in raw format (that is, as raw bytes). To convert these bytes into a useful object (like a JSON object, or a character vector, or a list), is highly recommended to apply the `httr::content()` function over this `content` element.

Although this being a useful output format (i.e. `'response'` object) (specially because it brings all of the available data), you might want a more "formatted" (or friendly) output. In this case, you can use the `.output_format` argument to get a different output format.

With `.output_format = "figma_document"`, `get_figma_file()` use `figma::as_figma_document()` to convert the `'response'` object into a Figma Document object (i.e. a object of class `figma_document`), and returns it as the output. This `figma_document` object, is a normal R list, with only the data of your Figma file (See documentation of `figma::as_figma_document()` for more details).

With `.output_format = "tibble"`, `get_figma_file()` will use `figma::as_tibble()` to parse the output from the API to fit into a `tibble::tibble()` object. If you use this output format, you can also use the `simplified` argument to control if document metadata should be present in the resulting tibble (See examples section).

By default, `simplified` is set to `TRUE`, so `get_figma_file()` outputs a tibble with all the objects data from your Figma file, and their corresponding canvas metadata. However, it does not include any metadata from the document per se.

In other words, with `simplified = TRUE` you get all the data of the objects from each canvas in your Figma file, but you do not get any metadata from the document. That is okay, because you usually do not need these informations.

But if you want them in the resulting tibble, pass `simplified = FALSE` to `get_figma_file()`. If you want just the document metadata (and not the canvas or objects data), you might want to use the `get_document_info()` function instead of `get_figma_file()` (See `get_document_info()` documentation for more details).

Value

By default, `get_figma_file()` do not parse the output from the API, and returns the raw response object produced by the `httr` HTTP methods (e.g. `httr::GET()`).

But you can change this behavior with `.output_format` argument. With `.output_format = "tibble"`, a `tibble::tibble()` object is returned. With `.output_format = "figma_document"`, a object of class `figma_document` is returned (See Details section for more information).

Be aware of possible HTTP errors

To get the data of your Figma file, the functions from `figma` package make a HTTP request to the Figma API. But this request can fail for a number of reasons, and if this does happen, `get_figma_file()` will use `report_http_error()` to raise an error and report to the user, what kind of error message the Figma API returned. See `vignette("http-errors")` for more details.

See Also

[as_tibble](#)
[as_figma_document](#)

Examples

```
## Not run:
library(figma)

file_key <- "hch8YlkgauIIZ9raDzjPvCz"
token <- "my figma token secret ..."

# Returns a `response` object:
```

```

result <- figma::get_figma_file(file_key, token)

# Returns a `tibble` object:
result <- figma::get_figma_file(
  file_key, token, .output_format = "tibble"
)

# Returns the same `tibble` object as before
# but, now, with all the metadata from the
# Figma document too:
result <- figma::get_figma_file(
  file_key, token,
  .output_format = "tibble",
  simplified = FALSE
)

# Returns a `figma_document` object:
result <- figma::get_figma_file(
  file_key, token, .output_format = "figma_document"
)

## End(Not run)

```

get_figma_page

Get data of a specific canvas/page in a Figma File from the API

Description

This function uses the `/v1/files/` endpoint of Figma API to get the data of an specific canvas/page (or a set of canvas/pages) from a Figma file, and fit it into a R object.

Usage

```

get_figma_page(
  file_key,
  token,
  node_ids,
  geometry = FALSE,
  .output_format = "response",
  ...
)

```

Arguments

<code>file_key</code>	A string with the key of the Figma File you want to get;
<code>token</code>	A string with your personal Figma token to authenticate in the API;
<code>node_ids</code>	A string with the node ID (or a vector of strings with node IDs);
<code>geometry</code>	A boolean value indicating if you want to export vector data. Defaults to FALSE;

```
.output_format The output format. Options are "response", "figma_document", "tibble".
Defaults to "response";
...           Further arguments that are passed to parse_response_object();
```

Details

With `'get_figma_file()'` you get data of all objects in all canvas/pages of your Figma file. But with `'get_figma_page()'` you get data of all objects drawn in a specific set of canvas/pages of your Figma file.

Every canvas/page in a Figma file, is identified by a node ID. You can easily get this ID from the URL that appears in your browser when you access this canvas/page on the Figma platform (See `vignette("figma")` for more details).

After you collected this node ID, give it to `node_id` argument as a string. If you want to collect data from more than one canvas/page of your Figma file, give a vector of node IDs to `node_id` argument.

Value

By default, `get_figma_page()` do not parse the output from the API, and returns the raw response object produced by the `httr` HTTP methods (e.g. `httr::GET()`).

But you can change this behavior with `.output_format` argument. With `.output_format = "tibble"`, a `tibble::tibble()` object is returned. With `.output_format = "figma_document"`, a object of class `figma_document` is returned (See Details section for more information).

Be aware of possible HTTP errors

To get the data of your Figma file, the functions from `figma` package make a HTTP request to the Figma API. But this request can fail for a number of reasons, and if this does happen, `get_figma_page()` will use `report_http_error()` to raise an error and report to the user, what kind of error message the Figma API returned. See `vignette("http-errors")` for more details.

Examples

```
## Not run:
library(figma)
file_key <- "hch8YlkgauIZ9raDzjPvCz"
token <- "my figma token secret ... "
node_id <- "0%3A1"
result <- figma::get_figma_page(
  file_key, token, node_id
)

## End(Not run)
```

parse_response_object *Parse the response data and fit it into a R object*

Description

This function is usually called by `get_figma_file()`, `get_document_info()` and `get_figma_page()`, to process the response object returned by the `httr` HTTP methods, such as `httr::GET()`.

Usage

```
parse_response_object(response, .output_format, ...)
```

Arguments

`response` The response object returned by the `httr` HTTP methods (e.g. `httr::GET()`);
`.output_format` A string with the name of the output format chosen by the user;
`...` Further arguments passed by the caller;

Details

The functions from `figma` package adopts the philosophy to give, by default, the most raw and unprocessed result possible to the user. Because of it, `parse_response_object()` is usually called with `.output_format = "response"`, which makes the function to just return the input as is.

This unprocessed and raw input gives all of the possible information to the user (which is good for debugging). But this information is usually in a very messy and not friendly format, which makes harder for data analysis and transformation pipelines.

The `.output_format` argument provide an option for the user to choose a more friendly format. As an example, with `.output_format = "tibble"`, `parse_response_object()` will call `figma::as_tibble()` to parse the data of the response object, and fit it into a `tibble::tibble()` object.

quarto_website *A representation of a Quarto Website home webpage*

Description

The `'quarto_website'` object contain the data of the homepage for a Quarto Website that was drawn in a Figma file. In other words, a homepage was drawn in Figma, and then, it was imported to R trough the `figma` package.

Usage

```
quarto_website
```

Format

An object of class `response`, produced by `httr` HTTP methods (e.g. `httr::GET()`);

Details

This Figma file contains a single page/canvas, and each HTML component is a separate object in the Figma file. The name of each object in the page/canvas correspond to the CSS selector used to style this HTML component in a real Quarto Website.

This is a interesting structure, because you can use the name and the attributes of each object to build custom CSS code, that maybe matches the style of a webpage.

Is worth mentioning, that the `quarto_website` object is a `httr::response` object returned by `figma::get_figma_file()`. This `httr::response` object is just a simple R list with class `response`. The elements of this list and their description are:

- `url`: The URL used in the HTTP request made to the Figma API;
- `status_code`: The HTTP status code returned by the Figma API;
- `headers` and `all_headers`: The list of headers returned by the Figma API;
- `cookies`: A named list of cookies returned by the Figma API;
- `content`: The body of the response, as raw vector. See `httr::content()` for various ways to parse this content;
- `date` and `times`: Timing information about the HTTP request made to the Figma API;
- `handle`: The handle associated with the url;
- `request`: The URL, HTTP method and options used in the HTTP request made to the Figma API;

All data of the Figma file is stored in the `content` element. However, this data is in raw format (i.e. in raw bytes). To convert these raw bytes into a useful format like a R list, or a string, you should use the `httr::content()` function. See `vignette("figma")` for a detailed description of the data present in this content element;

Examples

```
library(figma)
str(quarto_website)
```

select_any_of

Select any of the elements in a object

Description

Safely select elements of a named object (like a named R list).

Usage

```
select_any_of(x, elements)
```

Arguments

<code>x</code>	A object with names attribute (usually a named R list);
<code>elements</code>	A vector of strings with the name of the elements to search for;

Details

This functions works in a similar way to `dplyr::any_of()`. It tries to select any element of `x` that is in the vector given by the user in the `elements` argument.

In other words, if the user gives the vector `c("a", "c", "e")`, `select_any_of()` will search for elements "a", "c" and "e" in the `x` object, and will select any of these elements if it finds them.

But `dplyr::any_of()` is designed to work with columns of a `data.frame`, and `figma::select_any_of()` is designed to work specially with elements of a named list (although it can be used to select columns of a `data.frames` as well).

Value

A subset of the `x` if it finds any of the elements described in the `element` argument.

<code>untitled_file</code>	<i>An example of a simple Figma file</i>
----------------------------	--

Description

The `untitled_file` object is an example of what kind of object the functions from `figma` package tend to return to the user. This object was produced by a call to `figma::get_figma_file()`, and contains the data of a very simple Figma file entitled "Untitled".

Usage

```
untitled_file
```

Format

An object of class `response`, produced by `httr` HTTP methods (e.g. `httr::GET()`);

Details

This "Untitled" Figma file have two canvas/pages, called "Page 1" and "Page 2". There are three objects drawn in Page 1 (a `RECTANGLE`, a `TEXT` and a `VECTOR` object), and, there are two objects drawn in Page 2 (a `RECTANGLE` and a `TEXT` object).

The `untitled_file` object contains the `httr::response` object returned by `figma::get_figma_file()`. This `httr::response` object is just a simple R list with class `response`. The elements of this list and their description are:

- `url`: The URL used in the HTTP request made to the Figma API;
- `status_code`: The HTTP status code returned by the Figma API;

- `headers` and `all_headers`: The list of headers returned by the Figma API;
- `cookies`: A named list of cookies returned by the Figma API;
- `content`: The body of the response, as raw vector. See `httr::content()` for various ways to parse this content;
- `date` and `times`: Timing information about the HTTP request made to the Figma API;
- `handle`: The handle associated with the url;
- `request`: The URL, HTTP method and options used in the HTTP request made to the Figma API;

All data of the Figma file is stored in the `content` element. However, this data is in raw format (i.e. in raw bytes). To convert these raw bytes into a useful format like a R list, or a string, you should use the `httr::content()` function. See `vignette("figma")` for a detailed description of the data present in this content element;

Examples

```
library(figma)
str(untitled_file)
```


Index

* datasets

- default_attrs, 6
- document_attrs, 6
- quarto_website, 13
- untitled_file, 15

as_figma_document, 2, 10
as_tibble, 3, 10

build_query_string, 4
build_request_url, 5

default_attrs, 6
document_attrs, 6

get_document_info, 7
get_endpoint_url, 8
get_figma_file, 9
get_figma_page, 11

parse_response_object, 13

quarto_website, 13

select_any_of, 14

untitled_file, 15