Package 'gofigR'

March 19, 2025

Type Package

Title Client for 'GoFigr.io'

Version 0.3.1

Description Integrates with your 'RMarkdown' documents to automatically publish figures to the <https://GoFigr.io> service. Supports both 'knitr' and interactive execution within 'RStudio'.

License MIT + file LICENSE

Encoding UTF-8

URL https://github.com/GoFigr/gofigR

BugReports https://github.com/GoFigr/gofigR/issues

RoxygenNote 7.3.2

Imports httr, jsonlite, knitr, base64enc, readr, magick, qrcode, getPass, scriptName, rstudioapi, rsvg, cowplot, ggplotify

Suggests ggplot2, gplots

NeedsCompilation no

Author Maciej Pacula [cre, aut], Flagstaff Solutions, LLC [cph]

Maintainer Maciej Pacula <maciej@gofigr.io>

Repository CRAN

Date/Publication 2025-03-19 19:10:10 UTC

Contents

`authenticate_jwt` *Performs JWT authentication with username and password. Saves tokens in the GoFigr client.*

Description

Performs JWT authentication with username and password. Saves tokens in the GoFigr client.

Usage

```
authenticate_jwt(gf)
```

Arguments

`gf`	GoFigr client

Value

NA

`check_configured` *Checks whether GoFigr has been correctly configured.*

Description

Checks whether GoFigr has been correctly configured.

Usage

```
check_configured(response = warning)
```

Arguments

response	function to use to show the warning message if not configured. Default: warning.

Value

TRUE if configured, FALSE otherwise

CONFIG_PATH *Default path for the config file*

Description

Default path for the config file

Usage

```
CONFIG_PATH
```

Format

An object of class character of length 1.

---

create_analysis *Creates a new analysis*

Description

Creates a new analysis

Usage

```
create_analysis(gf, name, description = NULL, workspace = NULL)
```

Arguments

gf	GoFigr client
name	analysis name
description	analysis description
workspace	analysis will be created under this workspace. Can be a workspace object or an API ID.

Value

created analysis

`create_api_key` *Creates a new API key. This function will only succeed if using password authentication.*

Description

Creates a new API key. This function will only succeed if using password authentication.

Usage

```
create_api_key(gf, name)
```

Arguments

`gf`	GoFigr client. Must be using password authentication.
`name`	human-readable name of the API key to create, e.g. "John's laptop"

Value

response JSON. The "token" property will contain the API key if successful.

`create_figure` *Creates a new figure. The created figure will be blank and won't contain any revisions.*

Description

Creates a new figure. The created figure will be blank and won't contain any revisions.

Usage

```
create_figure(gf, analysis, name, description = NULL)
```

Arguments

`gf`	GoFigr object
`analysis`	analysis under which to create the figure
`name`	name for the new figure
`description`	description for the new figure

Value

created figure object

create_revision *Creates a new revision*

Description

Creates a new revision

Usage

```
create_revision(gf, figure, metadata = list(), data = NULL)
```

Arguments

gf	GoFigr client
figure	figure under which to create the revision
metadata	metadata for the revision, as named list
data	list of Data objects

Value

created revision object

create_workspace *Creates a new workspace*

Description

Creates a new workspace

Usage

```
create_workspace(gf, name, description = NULL)
```

Arguments

gf	GoFigr client
name	workspace name
description	workspace description

Value

created workspace object

DATA_TYPES *List of data types supported by GoFigr*

Description

List of data types supported by GoFigr

Usage

```
DATA_TYPES
```

Format

An object of class `list` of length 5.

Value

list where names are human-readable names of data types, and values are corresponding API types (strings).

default_if_null *Returns a default value if argument is null or empty*

Description

Returns a default value if argument is null or empty

Usage

```
default_if_null(x, default)
```

Arguments

`x`	argument
`default`	default value if x is null, NA or ""

Value

x if not null, NA or "", or the default value

enable *Enables GoFigr support.*

Description

Enables GoFigr support.

Usage

```
enable(
  auto_publish = FALSE,
  analysis_api_id = NULL,
  analysis_name = NULL,
  workspace = NULL,
  create_analysis = TRUE,
  analysis_description = NULL,
  watermark = QR_WATERMARK,
  verbose = FALSE,
  debug = FALSE,
  show = "watermark"
)
```

Arguments

auto_publish	will publish all plots automatically if TRUE. Note that setting this option will re-assign plot() and print() in the global environment.
analysis_api_id	Analysis API ID (if analysis_name is NULL)
analysis_name	Analysis name (if analysis_api_id is NULL)
workspace	API ID of the workspace
create_analysis	if TRUE and analysis_name does not exist, it will be automatically created
analysis_description	analysis description if creating a new analysis
watermark	watermark class to use, e.g. QR_WATERMARK, LINK_WATERMARK or NO_WATERMARK
verbose	whether to show verbose output
debug	whether to show debugging information
show	which figure to display in the document: original, watermark, or hide. Note that this setting \ only affects the display and doesn't change what gets published: e.g. even if you choose to display \ the original figure, the watermarked version will still be published to GoFigr.

Value

named list of GoFigr options

encode_raw_data *Converts a GoFigr data object into R primitives that can be converted to JSON, performing base64 encoding of binary data.*

Description

Converts a GoFigr data object into R primitives that can be converted to JSON, performing base64 encoding of binary data.

Usage

```
encode_raw_data(data)
```

Arguments

data	GoFigr data object

Value

encoded data object

Examples

```
data <- make_raw_data("test", "text", list(a=1), charToRaw("abcdefksjdfklsd"))
encode_raw_data(data)
```

find_analysis *Finds an analysis by name, optionally creating one if it doesn't exist.*

Description

Finds an analysis by name, optionally creating one if it doesn't exist.

Usage

```
find_analysis(gf, name, description = NULL, workspace = NULL, create = FALSE)
```

Arguments

gf	GoFigr client
name	name of the analysis to find
description	description of the analysis if it needs to be created
workspace	parent workspace
create	if TRUE and the analysis doesn't exist, it will be created; throws an error otherwise.

Value

analysis object

`find_figure` *Finds a figure by name*

Description

Finds a figure by name

Usage

```
find_figure(gf, analysis, name, description = NULL, create = FALSE)
```

Arguments

`gf`	GoFigr client
`analysis`	parent analysis
`name`	name of the figure to find
`description`	description of the figure if it needs to be created
`create`	if TRUE and the figure doesn't exist, it will be created; throws an error otherwise.

Value

figure object

`find_workspace` *Finds a workspace by name, optionally creating it if doesn't exist*

Description

Finds a workspace by name, optionally creating it if doesn't exist

Usage

```
find_workspace(gf, name, description = NULL, create = FALSE)
```

Arguments

`gf`	GoFigr client
`name`	name of the workspace to find
`description`	workspace description if creating a new one
`create`	if TRUE and workspace is not found, it will be created

Value

workspace if found; throws an error if not.

`get_analysis` *Fetches an analysis given an API ID.*

Description

Fetches an analysis given an API ID.

Usage

```
get_analysis(gf, api_id)
```

Arguments

`gf`	GoFigr client
`api_id`	API ID for the analysis

Value

analysis object

`get_api_id` *Returns obj$api_id if argument is an object, or identity if it's a string.*

Description

Returns obj$api_id if argument is an object, or identity if it's a string.

Usage

```
get_api_id(obj)
```

Arguments

`obj`	object for which to get the API ID

Value

API ID, a string

get_execution_context *Gets the execution context: input path, chunk code, and other metadata.*

Description

Gets the execution context: input path, chunk code, and other metadata.

Usage

```
get_execution_context()
```

Value

named list with the execution context.

Examples

```
get_execution_context()
```

get_figure *Fetches a figure given an API ID.*

Description

Fetches a figure given an API ID.

Usage

```
get_figure(gf, api_id)
```

Arguments

gf	GoFigr client
api_id	API ID for the figure

Value

figure object

get_options *Gets configured GoFigr options.*

Description

Gets configured GoFigr options.

Usage

```
get_options()
```

Value

GoFigr options, or NULL if not set.

get_revision *Fetches a revision given an API ID.*

Description

Fetches a revision given an API ID.

Usage

```
get_revision(gf, api_id)
```

Arguments

gf	GoFigr client
api_id	API ID for the revision

Value

revision object

get_revision_url *Gets the full URL for a revision*

Description

Gets the full URL for a revision

Usage

```
get_revision_url(rev)
```

Arguments

rev	revision object

Value

URL, a string

get_title *Gets a title from a plot*

Description

Gets a title from a plot

Usage

```
get_title(p)
```

Arguments

p	plot object

Value

title or NULL

`get_workspace` *Retrieves workspace details.*

Description

Retrieves workspace details.

Usage

```
get_workspace(gf, api_id)
```

Arguments

`gf`	GoFigr client
`api_id`	API ID of the workspace

Value

workspace details

`gfconfig` *Configures gofigr for use on this machine. Saves configuration to ~/.gofigr.*

Description

Configures gofigr for use on this machine. Saves configuration to ~/.gofigr.

Usage

```
gfconfig(max_attempts = 3)
```

Arguments

`max_attempts`	Maximum number of password attempts

Value

No return value

gf_plot *Plots and publishes an object (if supported)*

Description

Plots and publishes an object (if supported)

Usage

```
gf_plot(...)
```

Arguments

...	passed directly to plot

Value

result of the call to plot(...)

gf_print *Prints and publishes an object (if supported)*

Description

Prints and publishes an object (if supported)

Usage

```
gf_print(...)
```

Arguments

...	passed directly to print

Value

result of the call to print(...)

ggwatermark *Applies a watermark to a plot object/function.*

Description

Applies a watermark to a plot object/function.

Usage

```
ggwatermark(qr, plot_obj)
```

Arguments

qr	pre-generated QR code, as an image
plot_obj	plot object

Value

ggplot object with the watermark applied

gofigr_cat *Equivalent to cat but only outputs if GoFigr client is verbose.*

Description

Equivalent to cat but only outputs if GoFigr client is verbose.

Usage

```
gofigr_cat(gf, content, ...)
```

Arguments

gf	GoFigr client
content	text to print
...	passed to cat

Value

NA

gofigr_client *Creates and configures a GoFigr client. You can login either using a username & password or an API key. See examples.*

Description

Username, password, API key and workspace are read from the GoFigr configuration file (~/.gofigr) or environment variables if not supplied:

Usage

```
gofigr_client(
  username = NULL,
  password = NULL,
  api_key = NULL,
  url = NULL,
  anonymous = FALSE,
  verbose = FALSE,
  workspace = NULL,
  ignore_config = FALSE
)
```

Arguments

username	username (if not using API key)
password	password (if not using API key)
api_key	API key (if not using password authentication)
url	API URL (optional, you generally won't want to modify this)
anonymous	whether to login anonymously
verbose	set to TRUE to enable verbose output
workspace	default workspace (API ID)
ignore_config	if TRUE, will ignore environment variables and other external configuration

Details

* GF_USERNAME or config$username * GF_PASSWORD or config$password * GF_API_KEY or config$api_key * GF_WORKSPACE of config$workspace * GF_URL or config$url

Value

configured GoFigr client which you can pass to other functions

Examples

```
## Not run: gofigr_client()  # use config from ~/.gofigr or environment variables
## Not run: gofigr_client(username="joe", password="abc123") # password login
## Not run: gofigr_client(api_key="abcdef0123456789") # API key login
```

`gofigr_DELETE` *Wrapper for httr::DELETE that automatically handles authentication.*

Description

Wrapper for httr::DELETE that automatically handles authentication.

Usage

```
gofigr_DELETE(gf, url, expected_status_code = 200, ...)
```

Arguments

`gf`	configured GoFigr client
`url`	URL to make the request to, relative to the API URL e.g. user/
`expected_status_code`	expected HTTP response code. We will throw an exception if it differs.
`...`	passed to the httr request function

Value

result of calling the underlying httr request function

`gofigr_GET` *Wrapper for httr::GET that automatically handles authentication.*

Description

Wrapper for httr::GET that automatically handles authentication.

Usage

```
gofigr_GET(gf, url, expected_status_code = 200, ...)
```

Arguments

`gf`	configured GoFigr client
`url`	URL to make the request to, relative to the API URL e.g. user/
`expected_status_code`	expected HTTP response code. We will throw an exception if it differs.
`...`	passed to the httr request function

Value

result of calling the underlying httr request function

gofigr_make_handler *Wraps an HTTR method e.g. GET to provide relative URL resolution and authentication*

Description

Wraps an HTTR method e.g. GET to provide relative URL resolution and authentication

Usage

```
gofigr_make_handler(name, method)
```

Arguments

name	method name, e.g. "GET"
method	HTTR method, e.g. httr::GET

Value

wrapped method which takes a GoFigr client, a relative URL and an expected HTTP status code.

gofigr_PATCH *Wrapper for httr::PATCH that automatically handles authentication.*

Description

Wrapper for httr::PATCH that automatically handles authentication.

Usage

```
gofigr_PATCH(gf, url, expected_status_code = 200, ...)
```

Arguments

gf	configured GoFigr client
url	URL to make the request to, relative to the API URL e.g. user/
expected_status_code	expected HTTP response code. We will throw an exception if it differs.
...	passed to the httr request function

Value

result of calling the underlying httr request function

`gofigr_POST` *Wrapper for httr::POST that automatically handles authentication.*

Description

Wrapper for httr::POST that automatically handles authentication.

Usage

```
gofigr_POST(gf, url, expected_status_code = 200, ...)
```

Arguments

`gf`	configured GoFigr client
`url`	URL to make the request to, relative to the API URL e.g. user/
`expected_status_code`	expected HTTP response code. We will throw an exception if it differs.
`...`	passed to the httr request function

Value

result of calling the underlying httr request function

`gofigr_PUT` *Wrapper for httr::PUT that automatically handles authentication.*

Description

Wrapper for httr::PUT that automatically handles authentication.

Usage

```
gofigr_PUT(gf, url, expected_status_code = 200, ...)
```

Arguments

`gf`	configured GoFigr client
`url`	URL to make the request to, relative to the API URL e.g. user/
`expected_status_code`	expected HTTP response code. We will throw an exception if it differs.
`...`	passed to the httr request function

Value

result of calling the underlying httr request function

`infer_workspace` *Returns the argument if a valid workspace is passed, or the default workspace from the GoFigr client otherwise. Throws an error if both are NULL.*

Description

Returns the argument if a valid workspace is passed, or the default workspace from the GoFigr client otherwise. Throws an error if both are NULL.

Usage

```
infer_workspace(gf, workspace)
```

Arguments

`gf`	GoFigr client
`workspace`	workspace or NULL

Value

workspace object

`intercept` *Wraps a plotting function (e.g. plot) so that its output is intercepted by GoFigr.*

Description

Wraps a plotting function (e.g. plot) so that its output is intercepted by GoFigr.

Usage

```
intercept(plot_func)
```

Arguments

`plot_func`	function to intercept

Value

intercepted function

Examples

```
gf_plot <- intercept(base::plot)
```

is_expired_token *Returns True if the response indicates an expired JWT token*

Description

Returns True if the response indicates an expired JWT token

Usage

```
is_expired_token(res)
```

Arguments

res	httr response

Value

True if token expired

is_intercept_on *Checks whether GoFigr intercept is on*

Description

Checks whether GoFigr intercept is on

Usage

```
is_intercept_on()
```

Value

TRUE if intercept is on, FALSE otherwise

LINK_WATERMARK *Draws a watermark with just a GoFigr link*

Description

Draws a watermark with just a GoFigr link

Usage

```
LINK_WATERMARK(revision, image)
```

Arguments

revision	GoFigr revision object for which to generate a watermark
image	Magick image to which to add the watermark

Value

a function which you can pass to enable_knitr(watermark)

list_analyses *Lists analyses under a workspace.*

Description

Lists analyses under a workspace.

Usage

```
list_analyses(gf, workspace_id = NULL)
```

Arguments

gf	GoFigr client
workspace_id	API id of the workspace

Value

list of analyses

list_workspaces *List all workspaces available to the user.*

Description

List all workspaces available to the user.

Usage

```
list_workspaces(gf)
```

Arguments

gf	GoFigr client

Value

List of workspaces

login_with_api_key *Prompts the user for an API key or creates a new one*

Description

Prompts the user for an API key or creates a new one

Usage

```
login_with_api_key(gf, max_attempts)
```

Arguments

gf	Password-authenticated GoFigr client
max_attempts	Maximum number of login attempts before giving up

Value

API key, either supplied by the user or newly created

login_with_username *Prompts the user for username & password and logs into GoFigr*

Description

Prompts the user for username & password and logs into GoFigr

Usage

```
login_with_username(max_attempts)
```

Arguments

max_attempts	maximum number of login attempts before giving up

Value

GoFigr client

make_code_data *Creates a GoFigr data object storing source code*

Description

Creates a GoFigr data object storing source code

Usage

```
make_code_data(
  name,
  contents_or_file,
  language,
  format = "text",
  metadata = NULL
)
```

Arguments

name	name of this code object
contents_or_file	contents, a character string or file object
language	programming language, e.g. Python or R
format	not supported at the moment; please use the default
metadata	metadata associated with this object

Value

GoFigr data object

`make_file_data` *Creates a GoFigr data object storing file data*

Description

Creates a GoFigr data object storing file data

Usage

```
make_file_data(name, file_or_raw, path = NULL, metadata = NULL)
```

Arguments

`name`	name of this file
`file_or_raw`	image data, either a file or a raw vector
`path`	file path
`metadata`	metadata associated with this file

Value

GoFigr data object

`make_image_data` *Creates a GoFigr data object storing image data*

Description

Creates a GoFigr data object storing image data

Usage

```
make_image_data(name, file_or_raw, format, is_watermarked, metadata = NULL)
```

Arguments

`name`	name of this image
`file_or_raw`	image data, either a file or a raw vector
`format`	format, e.g. "png"
`is_watermarked`	whether this file has a GoFigr watermark
`metadata`	metadata associated with this image

Value

GoFigr data object

make_raw_data *Creates a GoFigr data object which can be attached to revisions.*

Description

Creates a GoFigr data object which can be attached to revisions.

Usage

```
make_raw_data(name, type, metadata, data)
```

Arguments

name	name of this data
type	data type, e.g. DATA_TYPES$image
metadata	metadata associated with this data object
data	raw bytes

Value

data object

make_table_data *Creates a GoFigr data object storing data.frame/tabular data*

Description

Creates a GoFigr data object storing data.frame/tabular data

Usage

```
make_table_data(name, frame, metadata = NULL)
```

Arguments

name	name of this data object
frame	data.frame
metadata	metadata associated with this data object

Value

GoFigr data object

make_text_data *Creates a GoFigr data object to store text*

Description

Creates a GoFigr data object to store text

Usage

```
make_text_data(name, contents, metadata = NULL)
```

Arguments

name	name of this data object
contents	contents, a character string
metadata	metadata associated with this object

Value

GoFigr data object

NO_WATERMARK *Does not draw any watermarks.*

Description

Does not draw any watermarks.

Usage

```
NO_WATERMARK
```

Format

An object of class NULL of length 0.

Value

does not return anything (NULL)

print.gofigr *Default print method for a GoFigr client.*

Description

Default print method for a GoFigr client.

Usage

```
## S3 method for class 'gofigr'
print(x, ...)
```

Arguments

x	GoFigr client
...	passed to cat

Value

NA

print.gofigrdata *Default print representation of GoFigr data objects.*

Description

Default print representation of GoFigr data objects.

Usage

```
## S3 method for class 'gofigrdata'
print(x, ...)
```

Arguments

x	object to print
...	passed to cat

Value

NA

publish *Publishes a figure to the GoFigr service.*

Description

Publishes a figure to the GoFigr service.

Usage

```
publish(
  plot_obj,
  figure_name = NULL,
  input_path = NULL,
  input_contents = NULL,
  chunk_code = NULL,
  image_formats = c("eps"),
  data = NULL,
  metadata = NULL,
  show = TRUE
)
```

Arguments

plot_obj	plot to publish
figure_name	name of the figure. If NULL, it will be inferred from the figure's title
input_path	path to the source file
input_contents	contents of the source file
chunk_code	chunk code, if running R markdown
image_formats	image formats to save
data	optional data to save with this figure. The data will be saved as RDS.
metadata	optional metadata
show	whether to display the figure after publication

Value

GoFigr revision object

publish_base *Captures output from grid graphics (ggplot2, lattice, ComplexHeatmap, etc.) and publishes it to GoFigr.*

Description

Captures output from grid graphics (ggplot2, lattice, ComplexHeatmap, etc.) and publishes it to GoFigr.

Usage

```
publish_base(expr, ...)
```

Arguments

expr	the expression to plot
...	passed through to publish()

Value

GoFigr Revision object

QR_WATERMARK *Draws a watermark with a GoFigr link and a QR code*

Description

Draws a watermark with a GoFigr link and a QR code

Usage

```
QR_WATERMARK(revision, image)
```

Arguments

revision	GoFigr revision object for which to generate a watermark
image	Magick image to which to add the watermark

Value

a function which you can pass to enable_knitr(watermark)

read_config *Reads the GoFigr configuration, prioritizing environment variables over the config file:*

Description

* GF_USERNAME or config["username"] * GF_PASSWORD or config["password"] * GF_API_KEY or config["api_key"] * GF_WORKSPACE or config["workspace"] * GF_URL or config["url"]

Usage

```
read_config(path = CONFIG_PATH)
```

Arguments

path	path to the config file, default ~/.gofigr

Value

parsed configuration or empty list if not available

read_prompt *Reads a prompt from stdin and performs optional validation*

Description

Reads a prompt from stdin and performs optional validation

Usage

```
read_prompt(prompt, validate = NULL, attempt = 1, max_attempts = 2)
```

Arguments

prompt	prompt, e.g. "Enter username: "
validate	function input => transformed input if input is valid or, error
attempt	current attempt at getting a valid input
max_attempts	maximum number of attempts

Value

input, or result of validate(input) if validate is supplied

`refresh_jwt` *Refreshes the JWT access token. Attempts re-authentication if refresh fails.*

Description

Refreshes the JWT access token. Attempts re-authentication if refresh fails.

Usage

```
refresh_jwt(gf)
```

Arguments

`gf`	GoFigr client.

Value

NA

`response_to_JSON` *Convenience function for parsing JSON from httr responses*

Description

Convenience function for parsing JSON from httr responses

Usage

```
response_to_JSON(response)
```

Arguments

`response`	httr response

Value

parsed JSON

set_options *Sets GoFigr options.*

Description

Sets GoFigr options.

Usage

```
set_options(options)
```

Arguments

options	New options that will replace existing options.

Value

NA

stack_horizontally *Stacks images horizontally, centering them vertically.*

Description

Stacks images horizontally, centering them vertically.

Usage

```
stack_horizontally(images)
```

Arguments

images	vector of images to stack

Value

composite image

stack_vertically *Stacks images vertically, centering them horizontally.*

Description

Stacks images vertically, centering them horizontally.

Usage

```
stack_vertically(images)
```

Arguments

images	vector of images to stack

Value

composite image

suppress *Suppresses any automatic GoFigr publication hooks.*

Description

Suppresses any automatic GoFigr publication hooks.

Usage

```
suppress(func)
```

Arguments

func	function in which to suppress intercepts

Value

the function with GoFigr supressed

update_revision_data *Updates data associated with a figure*

Description

Updates data associated with a figure

Usage

```
update_revision_data(gf, revision, new_data, silent = FALSE)
```

Arguments

gf	GoFigr client
revision	revision or its API ID for which to update the data
new_data	new data, as a list of GoFigrData objects (e.g. make_image_data or make_text_data)
silent	whether to generate an activity. Internal use only.

Value

updated revision

user_info *Fetches user details for the currently logged in user.*

Description

Fetches user details for the currently logged in user.

Usage

```
user_info(gf)
```

Arguments

gf	GoFigr client

Value

user details

watermark_generator *Makes a watermark generator. You can use the result with enable(watermark=...).*

Description

Makes a watermark generator. You can use the result with enable(watermark=...).

Usage

```
watermark_generator(
  show_qr = TRUE,
  qr_size_px = c(100, 100),
  link_size_px = c(500, 100),
  link_bg = "#ffffff",
  font_color = "#000000",
  font_size = 14,
  font = "mono",
  dynamic_size = TRUE
)
```

Arguments

show_qr	show QR code
qr_size_px	two-element vector specifying the width, height of the QR code
link_size_px	two-element vector specifying the width, height of the link
link_bg	background color for the link
font_color	font color for the link
font_size	font size for the link
font	font name or family, e.g. "mono"
dynamic_size	whether to automatically adjust the watermark size depending on the size of the current graphics device

Value

a function which you can pass to enable_knitr(watermark)

Index