Package 'gofigR'

May 8, 2026

Type Package

Title Client for 'GoFigr.io'

Version 1.1.3

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 7.3.3

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

Suggests ggplot2, gplots, rmarkdown, withr, purrr, devtools, uuid, rvest, stringr, torch, torchvision, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Maciej Pacula <maciej@gofigr.io>

Repository CRAN

Date/Publication 2025-11-26 17:20:01 UTC

Contents

`asset_linked_to_figure` *Creates an object representing a relationship between a figure and an asset.*

Description

Creates an object representing a relationship between a figure and an asset.

Usage

```
asset_linked_to_figure(figure_revision, asset_revision, use_type = "indirect")
```

Arguments

`figure_revision`	figure revision ID or object
`asset_revision`	asset revision ID or object
`use_type`	direct or indirect

Value

relationship object

`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

calc_checksum *Calculates a checksum for a file*

Description

Calculates a checksum for a file

Usage

```
calc_checksum(path)
```

Arguments

path	path to the file

Value

checksum, as a hex digest

cat.gofigr_revision *Default cat method for GoFigr revisions.*

Description

Default cat method for GoFigr revisions.

Usage

```
cat.gofigr_revision(x, ...)
```

Arguments

x	revision
...	passed to cat

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 to the config file*

Description

Default path to the config file

Usage

```
CONFIG_PATH
```

Format

An object of class `character` of length 1.

Value

file path

create_analysis *Create a new analysis within a workspace.*

Description

Analyses act as containers for figures, data and revisions. This helper creates a fresh analysis under the specified workspace.

Usage

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

Arguments

gf	GoFigr client.
name	Human-readable analysis name.
description	Optional longer description of the analysis.
workspace	Workspace under which the analysis will be created. Can be a workspace object or an API ID. If 'NULL', the client's default workspace is used.

Value

The created analysis object as returned by the API.

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_asset *Creates a new asset*

Description

Creates a new asset

Usage

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

Arguments

gf	GoFigr client
workspace	parent workspace
name	name of the asset
description	description of the asset

Value

asset object

create_asset_revision *Creates a new asset revision*

Description

Creates a new asset revision

Usage

```
create_asset_revision(gf, asset, metadata = list(), data = NULL)
```

Arguments

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

Value

created revision object

create_figure *Create a new figure under an analysis.*

Description

The newly created figure is initially blank and contains no revisions. Use the revision helpers to attach content once the figure has been created.

Usage

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

Arguments

gf	GoFigr client.
analysis	Parent analysis under which to create the figure. Can be an analysis object or an API ID.
name	Human-readable name for the new figure.
description	Optional longer description of the figure's purpose or contents.

Value

The created figure object as returned by the API.

create_ggsave_args *Creates a list of arguments for ggplot2::ggsave() with optional dimension and DPI parameters.*

Description

Creates a list of arguments for ggplot2::ggsave() with optional dimension and DPI parameters.

Usage

```
create_ggsave_args(
  filename,
  plot,
  width = NULL,
  height = NULL,
  units = "in",
  dpi = NULL
)
```

Arguments

filename	output filename
plot	plot object
width	width of the output image. If NULL, not included in arguments.
height	height of the output image. If NULL, not included in arguments.
units	units for width and height. If NULL, not included in arguments.
dpi	resolution of the output image. If NULL, not included in arguments.

Value

list of arguments suitable for do.call(ggplot2::ggsave, ...)

---

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 a named list
data	list of Data objects

Value

created revision object

create_workspace *Create a new workspace.*

Description

Workspaces are top-level containers for analyses and figures. This helper creates a new workspace owned by the current user.

Usage

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

Arguments

gf	GoFigr client.
name	Human-readable workspace name.
description	Optional longer description of the workspace.

Value

The created workspace object as returned by the API.

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

`delete_analysis` *Delete an analysis by API ID.*

Description

This permanently removes the analysis and its associated figures and revisions from the workspace.

Usage

```
delete_analysis(gf, api_id)
```

Arguments

`gf`	GoFigr client.
`api_id`	Character string with the API ID of the analysis to delete.

Value

Invisibly returns ‘NULL‘. An error is thrown if the deletion fails.

enable *Enables GoFigr in the current R/Rmd file.*

Description

Enables GoFigr in the current R/Rmd file.

Usage

```
enable(
  auto_publish = FALSE,
  analysis_api_id = NULL,
  analysis_name = NULL,
  workspace = NULL,
  workspace_name = NULL,
  create_analysis = TRUE,
  create_workspace = TRUE,
  analysis_description = NULL,
  workspace_description = NULL,
  watermark = QR_WATERMARK,
  verbose = FALSE,
  debug = FALSE,
  api_key = NULL,
  url = NULL,
  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
workspace_name	Workspace name (if workspace is NULL)
create_analysis	if TRUE and analysis_name does not exist, it will be automatically created
create_workspace	if TRUE and workspace_name does not exist, it will be automatically created
analysis_description	analysis description if creating a new analysis
workspace_description	workspace description if creating a new workspace

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
api_key	GoFigr API key
url	GoFigr API URL
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 *Find an analysis by name, optionally creating it.*

Description

Searches the analyses within a workspace by name and, optionally, creates a new analysis when no match is found.

Usage

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

Arguments

gf	GoFigr client.
name	Name of the analysis to find.
description	Optional description to assign if a new analysis is created.
workspace	Parent workspace (object or API ID). If 'NULL', the client's default workspace is used.
create	Logical; if 'TRUE' and the analysis does not exist, a new one is created. If 'FALSE', an error is thrown when no matching analysis is found.

Value

An analysis object corresponding to the matching (or newly created) analysis.

find_asset_by_name *Finds an asset by name*

Description

Finds an asset by name

Usage

```
find_asset_by_name(gf, name)
```

Arguments

gf	GoFigr client
name	name of the asset to search for

Value

list of matching assets, or an empty list if none found

`find_asset_revision_by_hash` *Finds all asset revisions with a matching hash digest*

Description

Finds all asset revisions with a matching hash digest

Usage

```
find_asset_revision_by_hash(gf, digest, hash_type = "blake3")
```

Arguments

`gf`	GoFigr client
`digest`	hex digest string
`hash_type`	digest type

Value

list of asset revisions, or empty list

`find_config` *Finds the .gofigr config file in current directory or any of the parent directories. If the file cannot be found, will also check CONFIG_PATH.*

Description

Finds the .gofigr config file in current directory or any of the parent directories. If the file cannot be found, will also check CONFIG_PATH.

Usage

```
find_config(start_dir = NULL)
```

Arguments

`start_dir`	top-level directory where to start looking. getwd() by default.

Value

path to .gofigr, or NULL if not found

find_figure *Find a figure by name within an analysis.*

Description

Searches the figures attached to a given analysis by name and optionally creates a new figure when no match is found. This is often the most convenient way to obtain a figure handle in scripts and notebooks.

Usage

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

Arguments

gf	GoFigr client.
analysis	Parent analysis object (or environment) in which to look for the figure.
name	Name of the figure to find.
description	Optional description to use if a new figure must be created.
create	Logical; if 'TRUE' and the figure doesn't exist, a new one is created. If 'FALSE', an error is thrown when no matching figure is found.

Value

A figure object corresponding to the matching (or newly created) figure.

find_workspace *Find a workspace by name, optionally creating it.*

Description

Searches the workspaces visible to the current user by name and, optionally, creates a new workspace when no match is found.

Usage

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

Arguments

gf	GoFigr client.
name	Name of the workspace to find.
description	Optional description to use if a new workspace is created.
create	Logical; if 'TRUE' and the workspace does not exist, a new one is created. If 'FALSE', an error is thrown when no matching workspace is found.

Value

A workspace object corresponding to the matching (or newly created) workspace.

`get_analysis` *Fetch an analysis by API ID.*

Description

Fetch an analysis by API ID.

Usage

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

Arguments

`gf`	GoFigr client.
`api_id`	Character string with the API ID of the analysis to fetch.

Value

An analysis object as returned by the API.

`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_asset *Fetches an asset given an API ID.*

Description

Fetches an asset given an API ID.

Usage

```
get_asset(gf, api_id)
```

Arguments

gf	GoFigr client
api_id	API ID for the asset

Value

asset object

get_asset_revision *Gets an asset revision given an API ID*

Description

Gets an asset revision given an API ID

Usage

```
get_asset_revision(gf, api_id)
```

Arguments

gf	GoFigr client
api_id	API ID for the revision

Value

asset revision object

get_client *Gets the currently configured GoFigr client*

Description

Gets the currently configured GoFigr client

Usage

```
get_client()
```

Value

GoFigr client

get_data *Retrieves a data object. Use in conjunction with get_revision or get_asset_revision, to retrieve the full data for a data object.*

Description

Retrieves a data object. Use in conjunction with get_revision or get_asset_revision, to retrieve the full data for a data object.

Usage

```
get_data(gf, api_id)
```

Arguments

gf	GoFigr client
api_id	API ID of the data object

Value

full data object

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 *Fetch a single figure by API ID.*

Description

This is a low-level helper that retrieves the raw figure object from the GoFigr API. It is typically used after you already know the figure's 'api_id', for example from an analysis or figure listing.

Usage

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

Arguments

gf	GoFigr client created by 'gofigr_client()'.
api_id	Character string with the API ID of the figure to fetch.

Value

A figure object as returned by the API (an environment), including metadata and revision information.

get_options *Gets configured GoFigr options.*

Description

Gets configured GoFigr options.

Usage

```
get_options()
```

Value

GoFigr options, or NULL if not set.

get_qr_png *Generates a QR code and converts it into an img element.*

Description

Generates a QR code and converts it into an img element.

Usage

```
get_qr_png(url, xres = 400, yres = 400, width = 100, height = 100)
```

Arguments

url	URL to generate the code for
xres	QR width, in pixels
yres	QR height, in pixels
width	width of the HTML img element
height	height of the HTML img element

Value

HTML string

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 *Retrieve workspace details.*

Description

Retrieve workspace details.

Usage

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

Arguments

gf	GoFigr client created by ‘gofigr_client()‘.
api_id	Character string with the API ID of the workspace to fetch.

Value

A workspace object as returned by the API, including metadata and lists of analyses and members.

gfconfig *Interactive configuration helper for the GoFigr R client.*

Description

Runs a simple text-based wizard that logs into GoFigr, generates or verifies an API key, lets the user choose a default workspace, and then writes a configuration file to '~/.gofigr'. This configuration is used by 'gofigr_client()' when explicit credentials are not provided.

Usage

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

Arguments

max_attempts	Maximum number of password/API key attempts before the wizard aborts with an error.

Value

Invisibly returns 'NULL'. The main effect is writing configuration to disk and printing progress messages.

gfContainer *Generates a div container for the GoFigr widget.*

Description

Generates a div container for the GoFigr widget.

Usage

```
gfContainer(..., jc = "center")
```

Arguments

...	passed to the div
jc	justify-content CSS value

Value

styled div element

gfPlot *Defines a GoFigr plot area.*

Description

Defines a GoFigr plot area.

Usage

```
gfPlot(id, ...)
```

Arguments

id	ID of this plot area
...	same as plotOutput

Value

HTML elements

gfPlotServer *Creates a Shiny component to handle plotting and publishing. Has to be paired with a gfPlot element in the UI.*

Description

Creates a Shiny component to handle plotting and publishing. Has to be paired with a gfPlot element in the UI.

Usage

```
gfPlotServer(
  id,
  expr,
  metadata = NULL,
  env = parent.frame(),
  figure_name = NULL,
  quoted = FALSE,
  base_graphics = FALSE
)
```

Arguments

id	id of the gfPlot element
expr	expression generating a plot
metadata	metadata to publish with the figure. You can pass the shiny input object to capture input values.
env	environment in which to evaluate the expression
figure_name	name of the figure to publish under. Inferred from figure's title if NULL.
quoted	whether the passed expression is quoted
base_graphics	whether the passed expression uses base graphics

Value

moduleServer

---

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` *Resolve the workspace argument, falling back to the client's default.*

Description

Returns the supplied workspace if present, otherwise the default workspace configured on the GoFigr client. Throws an error if neither is available.

Usage

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

Arguments

`gf`	GoFigr client.
`workspace`	Optional workspace object or API ID. If 'NULL', the client's default workspace is used.

Value

A workspace object or API ID suitable for passing to other helpers.

`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 *List analyses within a workspace.*

Description

This is a convenience wrapper around ‘get_workspace()‘ that returns only the analyses associated with a workspace, rather than the full workspace object.

Usage

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

Arguments

gf	GoFigr client.
workspace_id	Optional API ID of the workspace to inspect. If ‘NULL‘, the default workspace configured on the client (‘gf$workspace‘) is used.

Value

A list of analysis objects associated with the selected workspace.

list_workspaces *List all workspaces available to the authenticated user.*

Description

List all workspaces available to the authenticated user.

Usage

```
list_workspaces(gf)
```

Arguments

gf	GoFigr client.

Value

A list of workspace objects visible to the current user.

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

Description

Given a password-authenticated GoFigr client, this helper either accepts an existing API key entered by the user or creates a new API key via the API. The newly created key is associated with the authenticated user.

Usage

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

Arguments

gf	Password-authenticated GoFigr client created by ‘gofigr_client()‘.
max_attempts	Maximum number of attempts when validating a user- supplied API key.

Value

A character string containing a valid API key, either supplied by the user or newly created.

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

Description

This function interactively requests credentials, attempts authentication against the GoFigr API, and retries a limited number of times on failure. It is primarily used by ‘gfconfig()‘ and is not intended for scripted use.

Usage

```
login_with_username(max_attempts)
```

Arguments

max_attempts	Maximum number of login attempts before giving up.

Value

A configured GoFigr client object authenticated with username and password.

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

new_asset_revision_from_file *Creates a new asset revision from file.*

Description

Creates a new asset revision from file.

Usage

```
new_asset_revision_from_file(gf, workspace_id, path)
```

Arguments

gf	GoFigr client
workspace_id	parent workspace in case we have to create a brand new asset
path	path to file

Value

asset revision 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.gofigr_revision *Default print method for GoFigr revisions.*

Description

Default print method for GoFigr revisions.

Usage

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

Arguments

x	revision
...	passed to base::print

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,
  base_convert = TRUE,
  width = NULL,
  height = NULL,
  units = "in",
  dpi = NULL
)
```

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
base_convert	whether to try converting base graphics to grid graphics
width	width of the output image. If NULL, uses current device dimensions.
height	height of the output image. If NULL, uses current device dimensions.
units	units for width and height. Default is "in" (inches). Other options include "cm", "mm", "px".
dpi	resolution of the output image. If NULL, uses ggsave default (300).

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.csv *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read.csv(path, ...)
```

Arguments

path	path to the file
...	passed to utils::read.csv

Value

data frame

`read.csv2` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read.csv2(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to utils::read.csv

Value

data frame

`read.xlsx` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read.xlsx(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to openxlsx::read.xlsx

Value

data frame

`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 = NULL)
```

Arguments

`path`	path to the config file, default find_config()

Value

parsed configuration or empty list if not available

`read_csv` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read_csv(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to readr::read_csv

Value

data frame

`read_csv2` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read_csv2(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to readr::read_csv2

Value

data frame

`read_delim` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read_delim(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to readr::read_delim

Value

data frame

`read_prompt` *Reads a line from stdin with optional validation and retry logic.*

Description

This is a small helper used by the GoFigr configuration wizard to collect user input interactively. When a validation function is supplied, the input will be repeatedly requested until it passes validation or the maximum number of attempts is reached.

Usage

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

Arguments

`prompt`	Character string shown to the user, e.g. "Enter username: ".
`validate`	Optional function taking a single character argument and either returning a transformed value or throwing an error if the value is invalid.
`attempt`	Current attempt number (used internally for recursion).
`max_attempts`	Maximum number of attempts before giving up and throwing an error.

Value

The raw input string, or the result of ‘validate(input)‘ if a validation function is supplied.

`read_tsv` *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
read_tsv(path, ...)
```

Arguments

`path`	path to the file
`...`	passed to readr::read_tsv

Value

data frame

`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

By default, ‘jsonlite::fromJSON‘ returns nested named lists. For mutable objects that we want to update in-place (e.g. adding ‘figures‘ to an ‘analysis‘), it’s more convenient to work with environments, since they are passed by reference.

Usage

```
response_to_JSON(response)
```

Arguments

`response`	httr response

Details

This helper converts named lists to environments recursively, while keeping unnamed lists (arrays) as lists of elements. Atomic values are returned as-is.

Value

parsed JSON where JSON objects are represented as environments and JSON arrays as R lists.

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

sync_file — *Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.*

Description

Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.

Usage

```
sync_file(path)
```

Arguments

path	path to the file

Value

null

sync_workspace_asset — *Syncs a file with the GoFigr service*

Description

1. If we haven't seen this file before, creates a new asset and a new revision
2. If we have seen the file but haven't seen this revision, creates a new revision
3. If we have seen this revision, returns the existing revision

Usage

```
sync_workspace_asset(gf, workspace_id, path)
```

Arguments

gf	GoFigr client
workspace_id	parent workspace in case we have to create a brand new asset
path	path to file

Value

asset revision object

try_base2grob *Tries to convert expression to a grob, returning it unchanged if it fails.*

Description

Tries to convert expression to a grob, returning it unchanged if it fails.

Usage

```
try_base2grob(expr)
```

Arguments

expr	expression/object to convert

Value

grob if successful, expr if not

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, assets = list())
```

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.
assets	list of asset revision IDs to be assocaited with this revision

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)

---

with_isolated_devices *Executes an expression while isolating any new graphics devices it creates.*

Description

Executes an expression while isolating any new graphics devices it creates.

Usage

```
with_isolated_devices(expr)
```

Arguments

expr	The R expression to evaluate.

Value

result of evaluating expr

Index