

Package ‘gofigR’

July 31, 2025

Type Package

Title Client for 'GoFigr.io'

Version 1.1.2

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, 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 2025-07-31 19:32:04 UTC

Contents

asset_linked_to_figure	4
authenticate_jwt	4
calc_checksum	5
cat.gofigr_revision	5

check_configured	6
CONFIG_PATH	6
create_analysis	7
create_api_key	7
create_asset	8
create_asset_revision	8
create_figure	9
create_revision	9
create_workspace	10
DATA_TYPES	10
default_if_null	11
delete_analysis	11
enable	12
encode_raw_data	13
find_analysis	13
find_asset_by_name	14
find_asset_revision_by_hash	14
find_config	15
find_figure	15
find_workspace	16
get_analysis	17
get_api_id	17
get_asset	18
get_asset_revision	18
get_client	19
get_data	19
get_execution_context	20
get_figure	20
get_options	21
get_qr_png	21
get_revision	22
get_revision_url	22
get_title	23
get_workspace	23
gfconfig	24
gfContainer	24
gfPlot	25
gfPlotServer	25
gf_plot	26
gf_print	27
ggwatermark	27
gofigr_cat	28
gofigr_client	28
gofigr_DELETE	29
gofigr_GET	30
gofigr_make_handler	30
gofigr_PATCH	31
gofigr_POST	31

gofigr_PUT	32
infer_workspace	32
intercept	33
is_expired_token	33
is_intercept_on	34
LINK_WATERMARK	34
list_analyses	35
list_workspaces	35
login_with_api_key	36
login_with_username	36
make_code_data	37
make_file_data	37
make_image_data	38
make_raw_data	39
make_table_data	39
make_text_data	40
new_asset_revision_from_file	40
NO_WATERMARK	41
print.gofigr	41
print.gofigrdata	42
print.gofigr_revision	42
publish	43
publish_base	44
QR_WATERMARK	44
read.csv	45
read.csv2	45
read.xlsx	46
read_config	46
read_csv	47
read_csv2	47
read_delim	48
read_prompt	48
read_tsv	49
refresh_jwt	49
response_to_JSON	50
set_options	50
stack_horizontally	51
stack_vertically	51
suppress	52
sync_file	52
sync_workspace_asset	53
try_base2grob	53
update_revision_data	54
user_info	54
watermark_generator	55
with_isolated_devices	56

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	<i>Calculates a checksum for a file</i>
---------------	---

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	<i>Default cat method for GoFigr revisions.</i>
---------------------	---

Description

Default cat method for GoFigr revisions.

Usage

cat.gofigr_revision(x, ...)

Arguments

x	revision
...	passed to cat

Value

NA

check_configured	<i>Checks whether GoFigr has been correctly configured.</i>
------------------	---

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	<i>Default path to the config file</i>
-------------	--

Description

Default path to the config file

Usage

```
CONFIG_PATH
```

Format

An object of class character of length 1.

Value

file path

create_analysis	<i>Creates a new analysis</i>
-----------------	-------------------------------

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	<i>Creates a new API key. This function will only succeed if using password authentication.</i>
----------------	---

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	<i>Creates a new asset</i>
--------------	----------------------------

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	<i>Creates a new asset revision</i>
-----------------------	-------------------------------------

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	<i>Creates a new figure. The created figure will be blank and won't contain any revisions.</i>
---------------	--

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	<i>Creates a new revision</i>
-----------------	-------------------------------

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	<i>Creates a new workspace</i>
------------------	--------------------------------

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	<i>List of data types supported by GoFigr</i>
------------	---

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	Deletes an analysis given an API ID.
-----------------	--------------------------------------

Description

Deletes an analysis given an API ID.

Usage

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

Arguments

gf	GoFigr client
api_id	API ID for the analysis

Value

server response

enable	<i>Enables GoFigr in the current R/Rmd file.</i>
--------	--

Description

Enables GoFigr in the current R/Rmd file.

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,
  api_key = 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
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
api_key	GoFigr API key
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	<i>Converts a GoFigr data object into R primitives that can be converted to JSON, performing base64 encoding of binary data.</i>
-----------------	--

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("abcdefksjdfkl1sd"))
encode_raw_data(data)
```

find_analysis	<i>Finds an analysis by name, optionally creating one if it doesn't exist.</i>
---------------	--

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_asset_by_name	<i>Finds an asset by name</i>
--------------------	-------------------------------

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	<i>Finds all asset revisions with a matching hash digest</i>
-----------------------------	--

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	<i>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.</i>
-------------	---

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	<i>Finds a figure by name</i>
-------------	-------------------------------

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	<i>Finds a workspace by name, optionally creating it if doesn't exist</i>
----------------	---

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_asset	<i>Fetches an asset given an API ID.</i>
-----------	--

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	<i>Gets an asset revision given an API ID</i>
--------------------	---

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	<i>Gets the currently configured GoFigr client</i>
------------	--

Description

Gets the currently configured GoFigr client

Usage

```
get_client()
```

Value

GoFigr client

get_data	<i>Retrieves a data object. Use in conjunction with get_revision or get_asset_revision, to retrieve the full data for a data object.</i>
----------	--

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	<i>Gets the execution context: input path, chunk code, and other meta-data.</i>
-----------------------	---

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	<i>Fetches a figure given an API ID.</i>
------------	--

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	<i>Gets configured GoFigr options.</i>
-------------	--

Description

Gets configured GoFigr options.

Usage

```
get_options()
```

Value

GoFigr options, or NULL if not set.

get_qr_png	<i>Generates a QR code and converts it into an img element.</i>
------------	---

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	<i>Fetches a revision given an API ID.</i>
--------------	--

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	<i>Gets the full URL for a revision</i>
------------------	---

Description

Gets the full URL for a revision

Usage

```
get_revision_url(rev)
```

Arguments

rev	revision object
-----	-----------------

Value

URL, a string

get_title	<i>Gets a title from a plot</i>
-----------	---------------------------------

Description

Gets a title from a plot

Usage

```
get_title(p)
```

Arguments

p	plot object
---	-------------

Value

title or NULL

get_workspace	<i>Retrieves workspace details.</i>
---------------	-------------------------------------

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	<i>Configures gofigr for use on this machine. Saves configuration to ~/.gofigr.</i>
----------	---

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

gfContainer	<i>Generates a div container for the GoFigr widget.</i>
-------------	---

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	<i>Defines a GoFigr plot area.</i>
--------	------------------------------------

Description

Defines a GoFigr plot area.

Usage

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

Arguments

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

Value

HTML elements

gfPlotServer	<i>Creates a Shiny component to handle plotting and publishing. Has to be paired with a gfPlot element in the UI.</i>
--------------	---

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	<i>Plots and publishes an object (if supported)</i>
---------	---

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	<i>Prints and publishes an object (if supported)</i>
----------	--

Description

Prints and publishes an object (if supported)

Usage

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

Arguments

... passed directly to print

Value

result of the call to print(...)

ggwatermark	<i>Applies a watermark to a plot object/function.</i>
-------------	---

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	<i>Equivalent to cat but only outputs if GoFigr client is verbose.</i>
------------	--

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	<i>Creates and configures a GoFigr client. You can login either using a username & password or an API key. See examples.</i>
---------------	--

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 or 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	<i>Wrapper for httr::DELETE that automatically handles authentication.</i>
---------------	--

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 http::GET that automatically handles authentication.*

Description

Wrapper for http::GET that automatically handles authentication.

Usage

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

Arguments

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

Value

result of calling the underlying http request function

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

Description

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

Usage

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

Arguments

<code>name</code>	method name, e.g. "GET"
<code>method</code>	HTTP method, e.g. http::GET

Value

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

gofigr_PATCH	<i>Wrapper for http::PATCH that automatically handles authentication.</i>
--------------	---

Description

Wrapper for http::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 http request function

Value

result of calling the underlying http request function

gofigr_POST	<i>Wrapper for http::POST that automatically handles authentication.</i>
-------------	--

Description

Wrapper for http::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 http request function

Value

result of calling the underlying http request function

gofigr_PUT	<i>Wrapper for http::PUT that automatically handles authentication.</i>
------------	---

Description

Wrapper for http::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 http request function

Value

result of calling the underlying http request function

infer_workspace	<i>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.</i>
-----------------	--

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	<i>Wraps a plotting function (e.g. plot) so that its output is intercepted by GoFigr.</i>
-----------	---

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	<i>Returns True if the response indicates an expired JWT token</i>
------------------	--

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

<code>is_intercept_on</code>	<i>Checks whether GoFigr intercept is on</i>
------------------------------	--

Description

Checks whether GoFigr intercept is on

Usage

```
is_intercept_on()
```

Value

TRUE if intercept is on, FALSE otherwise

<code>LINK_WATERMARK</code>	<i>Draws a watermark with just a GoFigr link</i>
-----------------------------	--

Description

Draws a watermark with just a GoFigr link

Usage

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

Arguments

<code>revision</code>	GoFigr revision object for which to generate a watermark
<code>image</code>	Magick image to which to add the watermark

Value

a function which you can pass to `enable_knitr(watermark)`

list_analyses	<i>Lists analyses under a workspace.</i>
---------------	--

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	<i>List all workspaces available to the user.</i>
-----------------	---

Description

List all workspaces available to the user.

Usage

```
list_workspaces(gf)
```

Arguments

gf	GoFigr client
----	---------------

Value

List of workspaces

login_with_api_key	<i>Prompts the user for an API key or creates a new one</i>
--------------------	---

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	<i>Prompts the user for username & password and logs into GoFigr</i>
---------------------	--

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	<i>Creates a GoFigr data object storing source code</i>
----------------	---

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	<i>Creates a GoFigr data object storing file data</i>
----------------	---

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	<i>Creates a GoFigr data object storing image data</i>
-----------------	--

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	<i>Creates a GoFigr data object which can be attached to revisions.</i>
---------------	---

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	<i>Creates a GoFigr data object storing data.frame/tabular data</i>
-----------------	---

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	<i>Creates a GoFigr data object to store text</i>
----------------	---

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	<i>Creates a new asset revision from file.</i>
------------------------------	--

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	<i>Does not draw any watermarks.</i>
--------------	--------------------------------------

Description

Does not draw any watermarks.

Usage

NO_WATERMARK

Format

An object of class NULL of length 0.

Value

does not return anything (NULL)

<code>print.gofigr</code>	<i>Default print method for a GoFigr client.</i>
---------------------------	--

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	<i>Default print representation of GoFigr data objects.</i>
------------------	---

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

print.gofigr_revision	<i>Default print method for GoFigr revisions.</i>
-----------------------	---

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	<i>Publishes a figure to the GoFigr service.</i>
---------	--

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  
)
```

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

Value

GoFigr revision object

<code>publish_base</code>	<i>Captures output from grid graphics (ggplot2, lattice, ComplexHeatmap, etc.) and publishes it to GoFigr.</i>
---------------------------	--

Description

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

Usage

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

Arguments

<code>expr</code>	the expression to plot
<code>...</code>	passed through to <code>publish()</code>

Value

GoFigr Revision object

<code>QR_WATERMARK</code>	<i>Draws a watermark with a GoFigr link and a QR code</i>
---------------------------	---

Description

Draws a watermark with a GoFigr link and a QR code

Usage

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

Arguments

<code>revision</code>	GoFigr revision object for which to generate a watermark
<code>image</code>	Magick image to which to add the watermark

Value

a function which you can pass to `enable_knitr(watermark)`

read.csv	<i>Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.</i>
----------	---

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 <code>utils::read.csv</code>

Value

data frame

read.csv2	<i>Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.</i>
-----------	---

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 <code>utils::read.csv</code>

Value

data frame

read.xlsx	<i>Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.</i>
-----------	---

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	<i>Reads the GoFigr configuration, prioritizing environment variables over the config file:</i>
-------------	---

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

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

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

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	<i>Reads a prompt from stdin and performs optional validation</i>
-------------	---

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

read_tsv	<i>Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.</i>
----------	---

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	<i>Refreshes the JWT access token. Attempts re-authentication if refresh fails.</i>
-------------	---

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	<i>Convenience function for parsing JSON from httr responses</i>
------------------	--

Description

Convenience function for parsing JSON from httr responses

Usage

```
response_to_JSON(response)
```

Arguments

response	httr response
----------	---------------

Value

parsed JSON

set_options	<i>Sets GoFigr options.</i>
-------------	-----------------------------

Description

Sets GoFigr options.

Usage

```
set_options(options)
```

Arguments

options	New options that will replace existing options.
---------	---

Value

NA

stack_horizontally	<i>Stacks images horizontally, centering them vertically.</i>
--------------------	---

Description

Stacks images horizontally, centering them vertically.

Usage

```
stack_horizontally(images)
```

Arguments

images	vector of images to stack
--------	---------------------------

Value

composite image

stack_vertically	<i>Stacks images vertically, centering them horizontally.</i>
------------------	---

Description

Stacks images vertically, centering them horizontally.

Usage

```
stack_vertically(images)
```

Arguments

images	vector of images to stack
--------	---------------------------

Value

composite image

suppress	<i>Suppresses any automatic GoFigr publication hooks.</i>
----------	---

Description

Suppresses any automatic GoFigr publication hooks.

Usage

suppress(func)

Arguments

func	function in which to suppress intercepts
------	--

Value

the function with GoFigr suppressed

sync_file	<i>Syncs a file with the GoFigr service and stores a reference. The file will be associated with all figures published after this call.</i>
-----------	---

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	<i>Syncs a file with the GoFigr service</i>
----------------------	---

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	<i>Tries to convert expression to a grob, returning it unchanged if it fails.</i>
---------------	---

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	<i>Updates data associated with a figure</i>
----------------------	--

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 associated with this revision

Value

updated revision

user_info	<i>Fetches user details for the currently logged in user.</i>
-----------	---

Description

Fetches user details for the currently logged in user.

Usage

```
user_info(gf)
```

Arguments

gf	GoFigr client
----	---------------

Value

user details

watermark_generator	<i>Makes a watermark generator. You can use the result with enable(watermark=...).</i>
---------------------	--

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

* datasets

- CONFIG_PATH, 6
- DATA_TYPES, 10
- NO_WATERMARK, 41

- asset_linked_to_figure, 4
- authenticate_jwt, 4

- calc_checksum, 5
- cat.gofigr_revision, 5
- check_configured, 6
- CONFIG_PATH, 6
- create_analysis, 7
- create_api_key, 7
- create_asset, 8
- create_asset_revision, 8
- create_figure, 9
- create_revision, 9
- create_workspace, 10

- DATA_TYPES, 10
- default_if_null, 11
- delete_analysis, 11

- enable, 12
- encode_raw_data, 13

- find_analysis, 13
- find_asset_by_name, 14
- find_asset_revision_by_hash, 14
- find_config, 15
- find_figure, 15
- find_workspace, 16

- get_analysis, 17
- get_api_id, 17
- get_asset, 18
- get_asset_revision, 18
- get_client, 19
- get_data, 19
- get_execution_context, 20

- get_figure, 20
- get_options, 21
- get_qr_png, 21
- get_revision, 22
- get_revision_url, 22
- get_title, 23
- get_workspace, 23
- gf_plot, 26
- gf_print, 27
- gfconfig, 24
- gfContainer, 24
- gfPlot, 25
- gfPlotServer, 25
- ggwatermark, 27
- gofigr_cat, 28
- gofigr_client, 28
- gofigr_DELETE, 29
- gofigr_GET, 30
- gofigr_make_handler, 30
- gofigr_PATCH, 31
- gofigr_POST, 31
- gofigr_PUT, 32

- infer_workspace, 32
- intercept, 33
- is_expired_token, 33
- is_intercept_on, 34

- LINK_WATERMARK, 34
- list_analyses, 35
- list_workspaces, 35
- login_with_api_key, 36
- login_with_username, 36

- make_code_data, 37
- make_file_data, 37
- make_image_data, 38
- make_raw_data, 39
- make_table_data, 39
- make_text_data, 40

`new_asset_revision_from_file`, [40](#)
`NO_WATERMARK`, [41](#)

`print.gofigr`, [41](#)
`print.gofigr_revision`, [42](#)
`print.gofigrdata`, [42](#)
`publish`, [43](#)
`publish_base`, [44](#)

`QR_WATERMARK`, [44](#)

`read.csv`, [45](#)
`read.csv2`, [45](#)
`read.xlsx`, [46](#)
`read_config`, [46](#)
`read_csv`, [47](#)
`read_csv2`, [47](#)
`read_delim`, [48](#)
`read_prompt`, [48](#)
`read_tsv`, [49](#)
`refresh_jwt`, [49](#)
`response_to_JSON`, [50](#)

`set_options`, [50](#)
`stack_horizontally`, [51](#)
`stack_vertically`, [51](#)
`suppress`, [52](#)
`sync_file`, [52](#)
`sync_workspace_asset`, [53](#)

`try_base2grob`, [53](#)

`update_revision_data`, [54](#)
`user_info`, [54](#)

`watermark_generator`, [55](#)
`with_isolated_devices`, [56](#)