

Package: gofigR (via r-universe)

October 14, 2026

Type Package

Title Client for 'GoFigr.io'

Version 2.0.2

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 7.3.3

Imports http, jsonlite, knitr, base64enc, readr, magick, qrcode, digest, scriptName, rstudioapi, rsvg, cowplot, ggplotify, shiny, shinyjs, uuid, git2r

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

Config/testthat/edition 3

Config/pak/sysreqs cmake make libmagick++-dev gsfonts libgit2-dev libsvg2-dev libssh2-1-dev libuv1-dev libssl-dev libx11-dev zlib1g-dev

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

Date/Publication 2026-06-25 19:37:52 UTC

RemoteUrl <https://github.com/gofigr/gofigr>

RemoteRef HEAD

RemoteSha 1d0745dce8447b0ee5f89651ebe5726f730e9216

Contents

asset_linked_to_figure	4
authenticate_jwt	5
calc_checksum	5
cat.gofigr_revision	6
check_clean_room_size	6
check_configured	7
checkbox	7
CONFIG_PATH	8
create_analysis	8
create_api_key	9
create_asset	9
create_asset_revision	10
create_figure	10
create_ggsave_args	11
create_revision	11
create_revision_auto_assign	12
create_workspace	13
DATA_TYPES	14
default_if_null	14
delete_analysis	15
dropdown	15
enable	16
encode_raw_data	17
find_analysis	18
find_asset_by_name	18
find_asset_revision_by_hash	19
find_config	20
find_figure	20
find_workspace	21
get_analysis	21
get_api_id	22
get_app_short_url	22
get_app_url	23
get_asset	23
get_asset_revision	24
get_auth0_config	24
get_client	25
get_data	25
get_execution_context	26
get_figure	26
get_options	27
get_qr_png	27
get_revision	28
get_revision_url	28
get_title	29
get_workspace	29

gf_plot	30
gf_print	30
gfconfig	31
gfContainer	31
gfPlot	32
gfPlotServer	32
ggwatermark	33
gofigr_cat	34
gofigr_client	34
gofigr_DELETE	35
gofigr_GET	36
gofigr_make_handler	36
gofigr_PATCH	37
gofigr_POST	37
gofigr_PUT	38
infer_workspace	38
intercept	39
is_expired_token	40
is_intercept_on	40
is_serializable	41
LINK_WATERMARK	41
list_analyses	42
list_sessions	42
list_workspaces	43
login_with_api_key	43
login_with_device_flow	44
make_code_data	44
make_file_data	45
make_image_data	46
make_raw_data	46
make_table_data	47
make_text_data	47
new_asset_revision_from_file	48
NO_WATERMARK	48
param_to_shiny_input	49
print.gofigr	49
print.gofigr_revision	50
print.gofigrdata	50
publish	51
publish_base	52
QR_WATERMARK	52
read.csv	53
read.csv2	53
read.xlsx	54
read_config	54
read_csv	55
read_csv2	55
read_delim	56

read_prompt	56
read_tsv	57
refresh_jwt	57
reproducible	58
reserve_short_id_prefix	59
response_to_JSON	59
round_trip_params	60
run_reproducible_gadget	60
server_info	61
set_options	61
slider	62
stack_horizontally	62
stack_vertically	63
static	63
suppress	64
sync_file	64
sync_workspace_asset	65
text_input	65
try_base2grob	66
update_revision_data	66
user_info	67
validate_params	67
watermark_generator	68
with_isolated_devices	69
Index	70

asset_linked_to_figure	<i>Creates an object representing a relationship between a figure and an asset.</i>
------------------------	---

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	<i>Performs JWT authentication with username and password. Saves tokens in the GoFigr client.</i>
------------------	---

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	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_clean_room_size	Check total DataFrame size against MAX_CLEAN_ROOM_BYTES. Returns a list with \$ok (logical) and \$total_bytes.
-----------------------	--

Description

Check total DataFrame size against MAX_CLEAN_ROOM_BYTES. Returns a list with \$ok (logical) and \$total_bytes.

Usage

check_clean_room_size(descriptors)

Arguments

descriptors	named list of gf_param descriptors
-------------	------------------------------------

Value

list with ok and total_bytes fields

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

checkbox	<i>Creates a checkbox parameter for use in reproducible functions.</i>
----------	--

Description

Creates a checkbox parameter for use in [reproducible](#) functions.

Usage

```
checkbox(default)
```

Arguments

default	default logical value
---------	-----------------------

Value

a gf_param object

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>Create a new analysis within a workspace.</i>
-----------------	--

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	<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, analysis = NULL)
```

Arguments

gf	GoFigr client
workspace	parent workspace
name	name of the asset
description	description of the asset
analysis	optional analysis API ID to scope the asset under

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

<code>gf</code>	GoFigr client
<code>asset</code>	asset under which to create the revision
<code>metadata</code>	metadata for the revision, as a named list
<code>data</code>	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

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

Value

The created figure object as returned by the API.

create_ggsave_args	<i>Creates a list of arguments for ggplot2::ggsave() with optional dimension and DPI parameters.</i>
--------------------	--

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

Description

Creates a new revision

Usage

```
create_revision(
  gf,
  figure,
  metadata = list(),
  data = NULL,
  client_id = NULL,
  short_id = 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
client_id	optional client-generated UUID for idempotent creation. When provided, the server uses this as the revision's ID and returns 409 Conflict if a revision with the same client_id already exists.
short_id	optional short ID for compact QR codes. Generated via reserve_short_id_prefix and make_short_id .

Value

created revision object

```
create_revision_auto_assign
```

Creates a new revision via the auto-assign endpoint.

Description

The server creates a temporary figure, processes the image data, then uses AI to assign the revision to the correct figure. The revision will have is_processing=TRUE until assignment completes.

Usage

```
create_revision_auto_assign(
  gf,
  analysis,
  metadata = list(),
  data = NULL,
  client_id = NULL,
  short_id = NULL
)
```

Arguments

gf	GoFigr client
analysis	analysis under which to auto-assign the revision
metadata	metadata for the revision, as a named list
data	list of Data objects
client_id	optional client-generated UUID for idempotent creation
short_id	optional short ID for compact QR codes

Value

created revision object

create_workspace	<i>Create a new workspace.</i>
------------------	--------------------------------

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	<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	<i>Returns a default value if argument is null or empty</i>
-----------------	---

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.

dropdown	Creates a dropdown parameter for use in reproducible functions.
----------	---

Description

Creates a dropdown parameter for use in [reproducible](#) functions.

Usage

```
dropdown(default, choices)
```

Arguments

default	default value
choices	character vector of allowed values

Value

a gf_param object

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,
  workspace_name = NULL,
  create_analysis = TRUE,
  create_workspace = TRUE,
  analysis_description = NULL,
  workspace_description = NULL,
  watermark = QR_WATERMARK,
  verbose = FALSE,
  debug = FALSE,
  auto_assign = NULL,
  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
auto_assign	if TRUE, use AI-based auto-assignment of figures instead of explicit figure names. Requires AI to be enabled on the server. Can be overridden per-call in publish .
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

Note

Do not call `publish()` from forked child processes (e.g. inside `parallel::mclapply` or `future::plan(multicore)`). Forks inherit the short ID counter, which can produce duplicate IDs and server-side conflicts. Graphics devices also do not survive forks.

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("abcdefksjdfklsd"))
encode_raw_data(data)
```

find_analysis	<i>Find an analysis by name, optionally creating it.</i>
---------------	--

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

Description

Finds an asset by name

Usage

```
find_asset_by_name(gf, name, analysis = NULL)
```

Arguments

gf	GoFigr client
name	name of the asset to search for
analysis	optional analysis API ID to scope the search

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", analysis = NULL)
```

Arguments

gf	GoFigr client
digest	hex digest string
hash_type	digest type
analysis	optional analysis API ID to scope the search

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>Find a figure by name within an analysis.</i>
-------------	--

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	<i>Find a workspace by name, optionally creating it.</i>
----------------	--

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	<i>Fetch an analysis by API ID.</i>
--------------	-------------------------------------

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	<i>Returns obj\$api_id if argument is an object, or identity if it's a string.</i>
------------	--

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_app_short_url	<i>Resolves the short-redirect URL used for watermark/QR code links. Returns the bare "gofigr.io" domain (without app. prefix) so the resulting URLs and QR codes are as compact as possible.</i>
-------------------	---

Description

Resolves the short-redirect URL used for watermark/QR code links. Returns the bare "gofigr.io" domain (without app. prefix) so the resulting URLs and QR codes are as compact as possible.

Usage

```
get_app_short_url(client = NULL)
```

Arguments

client	GoFigr client object. If NULL, returns APP_SHORT_URL.
--------	---

Value

short app base URL string

get_app_url	<i>Resolves the app (frontend) URL from a GoFigr client's API base URL. Used for full app links (asset URLs, "View on GoFigr" links, etc). For watermark/QR code URLs, use get_app_short_url instead.</i>
-------------	---

Description

Resolves the app (frontend) URL from a GoFigr client's API base URL. Used for full app links (asset URLs, "View on GoFigr" links, etc). For watermark/QR code URLs, use [get_app_short_url](#) instead.

Usage

```
get_app_url(client = NULL)
```

Arguments

client	GoFigr client object. If NULL, returns APP_URL.
--------	---

Value

app base URL 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_auth0_config	<i>Fetches Auth0 configuration from the server's /info endpoint.</i>
------------------	--

Description

Fetches Auth0 configuration from the server's /info endpoint.

Usage

```
get_auth0_config(api_url)
```

Arguments

api_url	Base API URL (e.g. "https://api.gofigr.io")
---------	---

Value

A list with domain, client_id, and audience, or NULL if unavailable.

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>Fetch a single figure by API ID.</i>
------------	---

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	<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>Retrieve workspace details.</i>
---------------	------------------------------------

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.

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(...)

gfconfig	<i>Interactive configuration helper for the GoFigr R client.</i>
----------	--

Description

Runs a simple text-based wizard that authenticates via Auth0 Device Code flow, 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(url = NULL, max_attempts = 3)
```

Arguments

url	API URL. Default: <code>https://api.gofigr.io</code>
max_attempts	Maximum number of 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	<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

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 httr::PUT that automatically handles authentication.</i>
------------	---

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	<i>Resolve the workspace to use, falling back to the client's default.</i>
-----------------	--

Description

Resolves a concrete workspace in this order of precedence:

1. the explicit 'workspace' argument, if supplied;
2. a workspace matched (or created) by 'workspace_name', if supplied;
3. the default workspace configured on the GoFigr client;
4. the single workspace accessible to the client, when exactly one is visible. This mirrors the Python client and is what scoped API keys (e.g. on compute instances, whose credentials carry no workspace) rely on.

Throws an error if none of these resolve to a workspace.

Usage

```
infer_workspace(  
  gf,  
  workspace = NULL,  
  workspace_name = NULL,  
  create_workspace = FALSE,  
  workspace_description = NULL  
)
```

Arguments

gf	GoFigr client.
workspace	Optional workspace object or API ID. Takes precedence over all other arguments.
workspace_name	Optional workspace name to look up (or create, when ‘create_workspace’ is ‘TRUE’).
create_workspace	Logical; if ‘TRUE’ and ‘workspace_name’ does not match an existing workspace, a new one is created.
workspace_description	Optional description used when creating a workspace by name.

Value

A workspace API ID suitable for passing to other helpers.

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	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

is_serializable	<i>Check whether a value is serializable for clean room storage.</i>
-----------------	--

Description

Check whether a value is serializable for clean room storage.

Usage

```
is_serializable(value)
```

Arguments

value	the value to check
-------	--------------------

Value

TRUE if serializable, FALSE otherwise

LINK_WATERMARK	<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

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	<i>List analyses within a workspace.</i>
---------------	--

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 <code>'NULL'</code> , the default workspace configured on the client (<code>'gf\$workspace'</code>) is used.

Value

A list of analysis objects associated with the selected workspace.

list_sessions	<i>Lists all active GoFigr sessions.</i>
---------------	--

Description

Lists all active GoFigr sessions.

Usage

```
list_sessions(pretty = TRUE)
```

Arguments

pretty	if TRUE (default), returns a data frame summary; if FALSE, returns the raw named list of session option environments
--------	--

Value

data frame or named list of sessions

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

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

Description

Given an 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	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_device_flow`*Authenticates via Auth0 Device Authorization Flow.*

Description

Requests a device code, displays the verification URL, optionally opens the browser, and polls for authorization. Returns a GoFigr client authenticated with the Auth0 access token.

Usage

```
login_with_device_flow(api_url, auth0_config, max_attempts = 3)
```

Arguments

<code>api_url</code>	Base API URL
<code>auth0_config</code>	Auth0 configuration from <code>get_auth0_config()</code>
<code>max_attempts</code>	Not used (kept for API consistency)

Value

A configured GoFigr client authenticated with an Auth0 token.

`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	<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`

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, analysis_id = NULL)
```

Arguments

<code>gf</code>	GoFigr client
<code>workspace_id</code>	parent workspace in case we have to create a brand new asset
<code>path</code>	path to file
<code>analysis_id</code>	optional analysis API ID to scope the asset under

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`)

param_to_shiny_input	<i>Build a Shiny input widget from a gf_param descriptor.</i>
----------------------	---

Description

Build a Shiny input widget from a gf_param descriptor.

Usage

```
param_to_shiny_input(ns, name, desc)
```

Arguments

ns	Shiny namespace function
name	parameter name
desc	gf_param descriptor

Value

Shiny input element, or NULL for static params

print.gofigr	<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.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

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	<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,  
  width = NULL,  
  height = NULL,  
  units = "in",  
  dpi = NULL,  
  auto_assign = 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).
auto_assign	if TRUE, use AI-based auto-assignment instead of explicit figure names. If NULL (default), uses the value set in enable .

Value

GoFigr revision object

publish_base	<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

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

Value

GoFigr Revision object

QR_WATERMARK	<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

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)`

<code>read.csv</code>	<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

<code>path</code>	path to the file
<code>...</code>	passed to <code>utils::read.csv</code>

Value

data frame

<code>read.csv2</code>	<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

<code>path</code>	path to the file
<code>...</code>	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 line from stdin with optional validation and retry logic.</i>
-------------	--

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	<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

reproducible	<i>Execute a function in a clean room environment and publish with full reproducibility metadata.</i>
--------------	---

Description

The function's formal parameters should use [slider](#), [dropdown](#), [checkbox](#), [text_input](#), or [static](#) to declare parameter types. Call [publish](#) inside the function body to publish figures with clean room metadata attached.

Usage

```
reproducible(
  fn,
  packages = character(0),
  imports = list(),
  name = NULL,
  interactive = FALSE,
  viewer = NULL
)
```

Arguments

<code>fn</code>	a function whose formals use <code>gf_param</code> defaults
<code>packages</code>	character vector of package names, or named list with versions (e.g. <code>list(ggplot2 = "3.5.0")</code>). Package versions are resolved automatically if not provided.
<code>imports</code>	named list mapping aliases to package names (e.g. <code>list(plt = "ggplot2")</code>). Optional.
<code>name</code>	optional display name for the function (shown in the webapp)
<code>interactive</code>	if TRUE, launches a Shiny gadget with parameter widgets for interactive exploration. The user can adjust parameters and re-run the figure live, then click Publish to publish with clean room metadata.
<code>viewer</code>	Shiny viewer function for the interactive gadget. Only used when <code>interactive = TRUE</code> . Defaults to <code>shiny::dialogViewer</code> . Use <code>shiny::paneViewer()</code> for the RStudio Viewer pane or <code>shiny::browserViewer()</code> for an external browser window.

Value

whatever the function body returns (typically the result of `publish()`)

`reserve_short_id_prefix`*Reserve a short ID prefix from the GoFigr server.*

Description

The prefix can be combined with a sequential base62 index to generate compact, unique short IDs for figure revisions. Returns NULL if the server does not support short IDs.

Usage

```
reserve_short_id_prefix(gf)
```

Arguments

<code>gf</code>	GoFigr client
-----------------	---------------

Value

8-character alphanumeric prefix string, or NULL on failure

`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

<code>response</code>	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.

round_trip_params	<i>Round-trip parameters through serialization to ensure what the function sees matches what will be stored. DataFrames go through Parquet; other values go through JSON.</i>
-------------------	---

Description

Round-trip parameters through serialization to ensure what the function sees matches what will be stored. DataFrames go through Parquet; other values go through JSON.

Usage

```
round_trip_params(descriptors)
```

Arguments

descriptors named list of gf_param descriptors

Value

named list of round-tripped default values

run_reproducible_gadget	<i>Run the Shiny gadget for interactive clean room exploration.</i>
-------------------------	---

Description

Run the Shiny gadget for interactive clean room exploration.

Usage

```
run_reproducible_gadget(
  fn,
  descriptors,
  resolved_params,
  packages,
  clean_env,
  context,
  viewer = NULL
)
```

Arguments

fn	the user's function
descriptors	named list of gf_param descriptors
resolved_params	named list of default values
packages	package names
clean_env	the clean execution environment
context	the clean room context
viewer	Shiny viewer function, or NULL for default dialogViewer

server_info	<i>Returns cached server info from the /info/ endpoint.</i>
-------------	---

Description

Returns cached server info from the /info/ endpoint.

Usage

```
server_info(gf)
```

Arguments

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

Value

named list with server info, or empty list if the call fails

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

slider	<i>Creates a slider parameter for use in reproducible functions.</i>
--------	--

Description

Creates a slider parameter for use in [reproducible](#) functions.

Usage

```
slider(default, min = NULL, max = NULL, step = NULL)
```

Arguments

default	default value (integer or numeric)
min	minimum value
max	maximum value
step	step size

Value

a gf_param object

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

static	<i>Creates a static (read-only) parameter for use in reproducible functions.</i>
--------	--

Description

Use this for data frames and other non-interactive values.

Usage

static(default)

Arguments

default the value

Value

a gf_param object

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, analysis_id = NULL)
```

Arguments

gf	GoFigr client
workspace_id	parent workspace in case we have to create a brand new asset
path	path to file
analysis_id	optional analysis API ID to scope the asset under

Value

asset revision object

text_input	<i>Creates a text input parameter for use in reproducible functions.</i>
------------	--

Description

Creates a text input parameter for use in [reproducible](#) functions.

Usage

```
text_input(default)
```

Arguments

default	default character value
---------	-------------------------

Value

a gf_param 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(),
  is_clean_room = NULL
)
```

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
is_clean_room	if TRUE, marks the revision as a clean room 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

validate_params	<i>Validate that all parameters are serializable for clean room storage. Returns a list with \$valid (logical) and \$errors (character vector of messages).</i>
-----------------	---

Description

Validate that all parameters are serializable for clean room storage. Returns a list with \$valid (logical) and \$errors (character vector of messages).

Usage

validate_params(descriptors)

Arguments

descriptors named list of gf_param descriptors

Value

list with valid and errors fields

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, [8](#)
- DATA_TYPES, [14](#)
- NO_WATERMARK, [48](#)

- asset_linked_to_figure, [4](#)
- authenticate_jwt, [5](#)

- calc_checksum, [5](#)
- cat.gofigr_revision, [6](#)
- check_clean_room_size, [6](#)
- check_configured, [7](#)
- checkbox, [7](#), [58](#)
- CONFIG_PATH, [8](#)
- create_analysis, [8](#)
- create_api_key, [9](#)
- create_asset, [9](#)
- create_asset_revision, [10](#)
- create_figure, [10](#)
- create_ggsave_args, [11](#)
- create_revision, [11](#)
- create_revision_auto_assign, [12](#)
- create_workspace, [13](#)

- DATA_TYPES, [14](#)
- default_if_null, [14](#)
- delete_analysis, [15](#)
- dropdown, [15](#), [58](#)

- enable, [16](#), [52](#)
- encode_raw_data, [17](#)

- find_analysis, [18](#)
- find_asset_by_name, [18](#)
- find_asset_revision_by_hash, [19](#)
- find_config, [20](#)
- find_figure, [20](#)
- find_workspace, [21](#)

- get_analysis, [21](#)
- get_api_id, [22](#)

- get_app_short_url, [22](#), [23](#)
- get_app_url, [23](#)
- get_asset, [23](#)
- get_asset_revision, [24](#)
- get_auth0_config, [24](#)
- get_client, [25](#)
- get_data, [25](#)
- get_execution_context, [26](#)
- get_figure, [26](#)
- get_options, [27](#)
- get_qr_png, [27](#)
- get_revision, [28](#)
- get_revision_url, [28](#)
- get_title, [29](#)
- get_workspace, [29](#)
- gf_plot, [30](#)
- gf_print, [30](#)
- gfconfig, [31](#)
- gfContainer, [31](#)
- gfPlot, [32](#)
- gfPlotServer, [32](#)
- ggwatermark, [33](#)
- gofigr_cat, [34](#)
- gofigr_client, [34](#)
- gofigr_DELETE, [35](#)
- gofigr_GET, [36](#)
- gofigr_make_handler, [36](#)
- gofigr_PATCH, [37](#)
- gofigr_POST, [37](#)
- gofigr_PUT, [38](#)

- infer_workspace, [38](#)
- intercept, [39](#)
- is_expired_token, [40](#)
- is_intercept_on, [40](#)
- is_serializable, [41](#)

- LINK_WATERMARK, [41](#)
- list_analyses, [42](#)
- list_sessions, [42](#)

list_workspaces, 43
login_with_api_key, 43
login_with_device_flow, 44

make_code_data, 44
make_file_data, 45
make_image_data, 46
make_raw_data, 46
make_table_data, 47
make_text_data, 47

new_asset_revision_from_file, 48
NO_WATERMARK, 48

param_to_shiny_input, 49
print.gofigr, 49
print.gofigr_revision, 50
print.gofigrdata, 50
publish, 17, 51, 58
publish_base, 52

QR_WATERMARK, 52

read.csv, 53
read.csv2, 53
read.xlsx, 54
read_config, 54
read_csv, 55
read_csv2, 55
read_delim, 56
read_prompt, 56
read_tsv, 57
refresh_jwt, 57
reproducible, 7, 15, 58, 62, 63, 65
reserve_short_id_prefix, 12, 59
response_to_JSON, 59
round_trip_params, 60
run_reproducible_gadget, 60

server_info, 61
set_options, 61
slider, 58, 62
stack_horizontally, 62
stack_vertically, 63
static, 58, 63
suppress, 64
sync_file, 64
sync_workspace_asset, 65

text_input, 58, 65

try_base2grob, 66

update_revision_data, 66
user_info, 67

validate_params, 67

watermark_generator, 68
with_isolated_devices, 69