

Package: ladder (via r-universe)

September 1, 2024

Title Get on to the Slides

Version 0.0.0.9002

Description Create tables from within R directly on Google Slides presentations. Currently supports matrix, data.frame and 'flextable' objects.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports gargle, cli, flextable, httpuv, httr, rlang, checkmate

URL <https://www.r-ladder.com>

BugReports <https://github.com/igrave/ladder/issues>

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

RemoteUrl <https://github.com/igrave/ladder>

RemoteRef HEAD

RemoteSha 9e8b37a4d442742d78e1ffa48c1f17ba994126b

Contents

add_to_slides	2
add_to_slides.data.frame	3
add_to_slides.flextable	4
add_to_slides.matrix	5
choose_slides	6
get_object_ids	6
get_slide_ids	7
ladder_auth	8
ladder_auth_configure	10
ladder_deauth	12
ladder_has_token	13
ladder_token	13

ladder_user	14
presentations.pages.get	14
presentations.pages.getThumbnail	15
slides_url	16

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

add_to_slides	<i>Add Object to Slides</i>
---------------	-----------------------------

Description

Add Object to Slides

Usage

```
add_to_slides(object, presentation_id, on = NULL, object_id, ...)
```

Arguments

object	An object to add to slides
presentation_id	The id from the Slides presentation
on	The id or number of the slide to add object to
object_id	A unique id for the new object on the slides
...	Other arguments used in methods
overwrite	If TRUE and an object with object_id exists it will be deleted and replaced.

Value

A presentation object after updating

Examples

```
s <- choose_slides()
obj <- iris[1:5, ]
add_to_slides(obj, s, on = 1, object_id = "iris_table")
```

```
add_to_slides.data.frame
```

Add data frame to Slides

Description

Add data frame to Slides

Usage

```
## S3 method for class 'data.frame'
add_to_slides(
  object,
  presentation_id,
  on = NULL,
  object_id = new_id("table"),
  digits = NULL,
  overwrite = FALSE,
  ...
)
```

Arguments

<code>object</code>	A data.frame
<code>presentation_id</code>	The id from the Slides presentation
<code>on</code>	The id or number of the slide to add to
<code>object_id</code>	A unique id for the table
<code>digits</code>	the minimum number of significant digits, see format . If NULL <code>getOption("digits")</code> is used.
<code>overwrite</code>	If TRUE and an object with <code>object_id</code> exists it will be deleted and replaced.
<code>...</code>	Not used in this method

Details

The data frame is added as a table with the column names in bold as the first row. For other formatting use the `flextable` package and [add_to_slides.flextable](#).

Value

A presentation object after updating

Examples

```
s <- choose_slides()
obj <- iris[1:5, ]
add_to_slides(obj, s, on = 1)
```

add_to_slides.flextable

Add table to Slides

Description

Add table to Slides

Usage

```
## S3 method for class 'flextable'
add_to_slides(
  object,
  presentation_id,
  on = NULL,
  object_id = new_id("table"),
  overwrite = FALSE,
  ...
)
```

Arguments

object	A table object (flextable)
presentation_id	The id from the Slides presentation
on	The id or number of the slide to add to
object_id	A unique id for the table
overwrite	If TRUE and an object with object_id exists it will deleted and replaced.
...	Not used in this method

Value

A presentation object after updating

Examples

```
s <- choose_slides()
library(flextable)
ft <- flextable(iris[1:5, ])
ft <- theme_box(ft)
ft <- color(ft, i = 1:3, j = 1:2, "pink", part = "body")
ft <- autofit(ft)
add_to_slides(ft, s, on = 1)
```

add_to_slides.matrix *Add matrix to Slides*

Description

Add matrix to Slides

Usage

```
## S3 method for class 'matrix'
add_to_slides(
  object,
  presentation_id,
  on = NULL,
  object_id = new_id("table"),
  digits = NULL,
  overwrite = FALSE,
  ...
)
```

Arguments

object	A matrix
presentation_id	The id from the Slides presentation
on	The id or number of the slide to add to
object_id	A unique id for the table
digits	the minimum number of significant digits, see format . If NULL <code>getOption("digits")</code> is used.
overwrite	If TRUE and an object with <code>object_id</code> exists it will be deleted and replaced.
...	Not used in this method

Details

The matrix is added as a table without any row or column names.

Value

A presentation object is returned invisibly

Examples

```
s <- choose_slides()
obj <- cov(iris[, 1:4])
add_to_slides(obj, s, on = 1)
```

choose_slides	<i>Choose Slides presentation</i>
---------------	-----------------------------------

Description

Opens a webpage for a user to authenticate with Google and select a presentation. This presentation is then authorised for use with ladder.

Usage

```
choose_slides()
```

Value

A presentation id

Examples

```
id <- choose_slides()
```

get_object_ids	<i>Get ids of objects on Slides</i>
----------------	-------------------------------------

Description

Get ids of objects on Slides

Usage

```
get_object_ids(presentation_id)
```

Arguments

presentation_id
character, the presentation id

Value

A list of character vectors of object ids. The list has elements for each page. If a slide page has no objects the list element is NULL otherwise a character vector containing all object ids on that page. Contains ids for all tables, images, lines, shapes, etc.

Examples

```
s <- choose_slides()  
get_object_ids(s)
```

get_slide_ids *Get ids of Slides pages*

Description

Get ids of Slides pages

Usage

```
get_slide_ids(presentation_id)
```

Arguments

presentation_id
character, the presentation id

Value

A vector of slide ids.

Examples

```
s <- choose_slides()  
get_slide_ids(s)
```

ladder_auth

*Authorize ladder***Description**

Authorize ladder to view and manage your presentations. This function is a wrapper around [gargle::token_fetch\(\)](#).

By default, you are directed to a web browser, asked to sign in to your Google account, and to grant ladder permission to operate on your behalf with Google Slides. By default, with your permission, these user credentials are cached in a folder below your home directory, from where they can be automatically refreshed, as necessary. Storage at the user level means the same token can be used across multiple projects and tokens are less likely to be synced to the cloud by accident.

Usage

```
ladder_auth(
  email = gargle::gargle_oauth_email(),
  path = NULL,
  subject = NULL,
  scopes = c("https://www.googleapis.com/auth/drive.file",
            "https://www.googleapis.com/auth/presentations.currentonly"),
  cache = gargle::gargle_oauth_cache(),
  use_oob = gargle::gargle_oob_default(),
  token = NULL
)
```

Arguments

email Optional. If specified, email can take several different forms:

- "jane@gmail.com", i.e. an actual email address. This allows the user to target a specific Google identity. If specified, this is used for token lookup, i.e. to determine if a suitable token is already available in the cache. If no such token is found, email is used to pre-select the targeted Google identity in the OAuth chooser. (Note, however, that the email associated with a token when it's cached is always determined from the token itself, never from this argument).
- "*@example.com", i.e. a domain-only glob pattern. This can be helpful if you need code that "just works" for both alice@example.com and bob@example.com.
- TRUE means that you are approving email auto-discovery. If exactly one matching token is found in the cache, it will be used.
- FALSE or NA mean that you want to ignore the token cache and force a new OAuth dance in the browser.

Defaults to the option named "gargle_oauth_email", retrieved by [gargle_oauth_email\(\)](#) (unless a wrapper package implements different default behavior).

path	JSON identifying the service account, in one of the forms supported for the <code>txt</code> argument of <code>jsonlite::fromJSON()</code> (typically, a file path or JSON string).
subject	An optional subject claim. Specify this if you wish to use the service account represented by <code>path</code> to impersonate the subject, who is a normal user. Before this can work, an administrator must grant the service account domain-wide authority. Identify the user to impersonate via their email, e.g. <code>subject = "user@example.com"</code> . Note that <code>gargle</code> automatically adds the non-sensitive <code>"https://www.googleapis.com/auth/userinfo.email"</code> scope, so this scope must be enabled for the service account, along with any other scopes being requested.
scopes	A character vector of scopes to request. Pick from those listed at https://developers.google.com/identity/protocols/oauth2/scopes .
cache	Specifies the OAuth token cache. Defaults to the option named <code>"gargle_oauth_cache"</code> , retrieved via <code>gargle_oauth_cache()</code> .
use_oob	Whether to use out-of-band authentication (or, perhaps, a variant implemented by <code>gargle</code> and known as "pseudo-OOB") when first acquiring the token. Defaults to the value returned by <code>gargle_oob_default()</code> . Note that (pseudo-)OOB auth only affects the initial OAuth dance. If we retrieve (and possibly refresh) a cached token, <code>use_oob</code> has no effect. If the OAuth client is provided implicitly by a wrapper package, its type probably defaults to the value returned by <code>gargle_oauth_client_type()</code> . You can take control of the client type by setting <code>options(gargle_oauth_client_type = "web")</code> or <code>options(gargle_oauth_client_type = "installed")</code> .
token	A token with class <code>Token2.0</code> or an object of <code>httr</code> 's class <code>request</code> , i.e. a token that has been prepared with <code>httr::config()</code> and has a <code>Token2.0</code> in the <code>auth_token</code> component.

Details

Most users, most of the time, do not need to call `ladder_auth()` explicitly – it is triggered by the first action that requires authorization. Even when called, the default arguments often suffice.

However, when necessary, `ladder_auth()` allows the user to explicitly:

- Declare which Google identity to use, via an email specification.
- Use a service account token or workload identity federation via `path`.
- Bring your own token.
- Customize scopes.
- Use a non-default cache folder or turn caching off.
- Explicitly request out-of-band (OOB) auth via `use_oob`.

If you are interacting with R within a browser (applies to RStudio Server, Posit Workbench, Posit Cloud, and Google Colaboratory), you need OOB auth or the pseudo-OOB variant. If this does not happen automatically, you can request it explicitly with `use_oob = TRUE` or, more persistently, by setting an option via `options(gargle_oob_default = TRUE)`.

The choice between conventional OOB or pseudo-OOB auth is determined by the type of OAuth client. If the client is of the "installed" type, `use_oob = TRUE` results in conventional OOB auth. If

the client is of the "web" type, `use_oob = TRUE` results in pseudo-OOB auth. Packages that provide a built-in OAuth client can usually detect which type of client to use. But if you need to set this explicitly, use the "gargle_oauth_client_type" option:

```
options(gargle_oauth_client_type = "web")      # pseudo-OOB
# or, alternatively
options(gargle_oauth_client_type = "installed") # conventional OOB
```

For details on the many ways to find a token, see `gargle::token_fetch()`. For deeper control over auth, use `ladder_auth_configure()` to bring your own OAuth client or API key. To learn more about gargle options, see `gargle::gargle_options`.

See Also

Other auth functions: `ladder_auth_configure()`, `ladder_deauth()`

Examples

```
# load/refresh existing credentials, if available
# otherwise, go to browser for authentication and authorization
ladder_auth()

# see user associated with current token
ladder_user()

# force use of a token associated with a specific email
ladder_auth(email = "jenny@example.com")
ladder_user()

# force the OAuth web dance
ladder_auth(email = NA)

# use a 'read only' scope, so it's impossible to edit or delete files
ladder_auth(scopes = "slides.readonly")

# use a service account token
ladder_auth(path = "foofy-83ee9e7c9c48.json")
```

`ladder_auth_configure` *Edit and view auth configuration*

Description

These functions give more control over and visibility into the auth configuration than `ladder_auth()` does. `ladder_auth_configure()` lets the user specify their own:

- OAuth client, which is used when obtaining a user token.

- API key. If ladder is de-authorized via `ladder_deauth()`, all requests are sent with an API key in lieu of a token.

See the vignette("get-api-credentials", package = "gargle") for more. If the user does not configure these settings, internal defaults are used.

`ladder_oauth_client()` and `ladder_api_key()` retrieve the currently configured OAuth client and API key, respectively.

Usage

```
ladder_auth_configure(client, path, api_key, app)
```

```
ladder_api_key()
```

```
ladder_oauth_client()
```

Arguments

<code>client</code>	A Google OAuth client, presumably constructed via <code>gargle::gargle_oauth_client_from_json()</code> . Note, however, that it is preferred to specify the client with JSON, using the <code>path</code> argument.
<code>path</code>	JSON downloaded from Google Cloud Console , containing a client id and secret, in one of the forms supported for the <code>txt</code> argument of <code>jsonlite::fromJSON()</code> (typically, a file path or JSON string).
<code>api_key</code>	API key.
<code>app</code>	[Deprecated] Replaced by the <code>client</code> argument.

Value

- `ladder_auth_configure()`: An object of R6 class `gargle::AuthState`, invisibly.
- `ladder_oauth_client()`: the current user-configured OAuth client.
- `ladder_api_key()`: the current user-configured API key.

See Also

Other auth functions: `ladder_auth()`, `ladder_deauth()`

Examples

```
# see and store the current user-configured OAuth client (probably `NULL`)
(original_client <- ladder_oauth_client())

# see and store the current user-configured API key (probably `NULL`)
(original_api_key <- ladder_api_key())

# the preferred way to configure your own client is via a JSON file
# downloaded from Google Developers Console
# this example JSON is indicative, but fake
path_to_json <- system.file(
```

```
    "extdata", "client_secret_installed.googleusercontent.com.json",
    package = "gargle"
)
ladder_auth_configure(path = path_to_json)

# this is also obviously a fake API key
ladder_auth_configure(api_key = "the_key_I_got_for_a_google_API")

# confirm the changes
ladder_oauth_client()
ladder_api_key()

# restore original auth config
ladder_auth_configure(client = original_client, api_key = original_api_key)
```

ladder_deauth

Suspend authorization

Description

Put ladder into a de-authorized state. Instead of sending a token, ladder will send an API key. This can be used to access public resources for which no Google sign-in is required. This is handy for using ladder in a non-interactive setting to make requests that do not require a token. It will prevent the attempt to obtain a token interactively in the browser. The user can configure their own API key via [ladder_auth_configure\(\)](#) and retrieve that key via [ladder_api_key\(\)](#). In the absence of a user-configured key, a built-in default key is used.

Usage

```
ladder_deauth()
```

See Also

Other auth functions: [ladder_auth\(\)](#), [ladder_auth_configure\(\)](#)

Examples

```
## Not run:
ladder_deauth()
ladder_user()

## End(Not run)
```

ladder_has_token	<i>Is there a token on hand?</i>
------------------	----------------------------------

Description

Reports whether ladder has stored a token, ready for use in downstream requests.

Usage

```
ladder_has_token()
```

Value

Logical.

See Also

Other low-level API functions: [ladder_token\(\)](#)

Examples

```
ladder_has_token()
```

ladder_token	<i>Produce configured token</i>
--------------	---------------------------------

Description

For internal use or for those programming around the Slides API. Returns a token pre-processed with [httr::config\(\)](#). Most users do not need to handle tokens "by hand" or, even if they need some control, [ladder_auth\(\)](#) is what they need. If there is no current token, [ladder_auth\(\)](#) is called to either load from cache or initiate OAuth2.0 flow. If auth has been deactivated via [ladder_deauth\(\)](#), [ladder_token\(\)](#) returns NULL.

Usage

```
ladder_token()
```

Value

A request object (an S3 class provided by [httr](#)).

See Also

Other low-level API functions: [ladder_has_token\(\)](#)

Examples

```
## Not run:  
ladder_token()  
  
## End(Not run)
```

ladder_user	<i>Get info on current user</i>
-------------	---------------------------------

Description

Reveals the email address of the user associated with the current token. If no token has been loaded yet, this function does not initiate auth.

Usage

```
ladder_user()
```

Value

An email address or, if no token has been loaded, NULL.

See Also

[gargle::token_userinfo\(\)](#), [gargle::token_email\(\)](#), [gargle::token_tokeninfo\(\)](#)

Examples

```
## Not run:  
ladder_user()  
  
## End(Not run)
```

presentations.pages.get	<i>Gets the latest version of the specified page in the presentation.</i>
-------------------------	---

Description

Gets the latest version of the specified page in the presentation.

Usage

```
presentations.pages.get(presentationId, pageObjectId)
```

Arguments

presentationId The ID of the presentation to retrieve
pageObjectId The object ID of the page to retrieve

See Also

[Google Documentation](#)

`presentations.pages.getThumbnail`

Generates a thumbnail of the latest version of the specified page.

Description

Generates a thumbnail of the latest version of the specified page in the presentation and returns a URL to the thumbnail image. This request counts as an [expensive read request](#) for quota purposes.

Usage

```
presentations.pages.getThumbnail(  
  presentationId,  
  pageObjectId,  
  thumbnailProperties.thumbnailSize = NULL,  
  thumbnailProperties.mimeType = NULL  
)
```

Arguments

presentationId The ID of the presentation to retrieve
pageObjectId The object ID of the page whose thumbnail to retrieve
thumbnailProperties.thumbnailSize
The optional thumbnail image size
thumbnailProperties.mimeType
The optional mime type of the thumbnail image

See Also

[Google Documentation](#)

slides_url	<i>Print presentation URL</i>
------------	-------------------------------

Description

Print presentation URL

Usage

```
slides_url(presentation_id, slide_id = NULL)
```

Arguments

presentation_id	ID of presentation
slide_id	Optional slide id to link directly to a certain slide. See get_slide_ids .

Value

Prints URL as a link and invisibly returns URL.

Examples

```
slides_url("example_id_won't_work_1234567asdfbg")  
slides_url("example_id_won't_work_1234567asdfbg", slide_id = "p")
```

Index

* auth functions

ladder_auth, 8
ladder_auth_configure, 10
ladder_deauth, 12

* low-level API functions

ladder_has_token, 13
ladder_token, 13

add_to_slides, 2

add_to_slides.data.frame, 3

add_to_slides.flextable, 3, 4

add_to_slides.matrix, 5

choose_slides, 6

format, 3, 5

gargle::AuthState, 11

gargle::gargle_oauth_client_from_json(),
11

gargle::gargle_options, 10

gargle::token_email(), 14

gargle::token_fetch(), 8, 10

gargle::token_tokeninfo(), 14

gargle::token_userinfo(), 14

gargle_oauth_cache(), 9

gargle_oauth_client_type(), 9

gargle_oauth_email(), 8

gargle_oob_default(), 9

get_object_ids, 6

get_slide_ids, 7, 16

htr, 13

htr::config(), 9, 13

jsonlite::fromJSON(), 9, 11

ladder_api_key (ladder_auth_configure),
10

ladder_api_key(), 12

ladder_auth, 8, 11, 12

ladder_auth(), 10, 13

ladder_auth_configure, 10, 10, 12

ladder_auth_configure(), 10, 12

ladder_deauth, 10, 11, 12

ladder_deauth(), 11, 13

ladder_has_token, 13, 13

ladder_oauth_client
(ladder_auth_configure), 10

ladder_token, 13, 13

ladder_user, 14

presentations.pages.get, 14

presentations.pages.getThumbnail, 15

slides_url, 16

Token2.0, 9