

# Package ‘rgoogleclassroom’

August 9, 2023

**Title** API Wrapper for Google Classroom and Google Forms

**Version** 0.9.1

**Description** This is a Google Forms and Google Classroom API Wrapper for R for managing Google Classrooms from R. The documentation for these APIs is here <<https://developers.google.com/forms/api/guides>> .

**License** GPL-3

**URL** <https://github.com/datatrail-jhu/rgoogleclassroom>

**BugReports** <https://github.com/datatrail-jhu/rgoogleclassroom/issues>

**Imports** attempt, curl, httr, jsonlite, purrr, R6, assertthat, lubridate, magrittr, openssl, ottrpal, stringr, dplyr, knitr,

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Candace Savonen [cre, aut]

**Maintainer** Candace Savonen <[csavonen@fredhutch.org](mailto:csavonen@fredhutch.org)>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2023-08-09 13:20:02 UTC

## R topics documented:

archive_course . . . . .	2
authorize . . . . .	3
auth_from_secret . . . . .	3
commit_to_form . . . . .	4
copy_form . . . . .	5

create_course . . . . .	5
create_coursework . . . . .	6
create_form . . . . .	7
create_material . . . . .	8
create_multiple_choice_question . . . . .	9
create_quiz . . . . .	10
create_text_question . . . . .	11
create_topic . . . . .	12
date_handler . . . . .	12
delete_course . . . . .	13
delete_coursework . . . . .	13
get_coursework_list . . . . .	14
get_coursework_properties . . . . .	14
get_course_list . . . . .	15
get_course_properties . . . . .	15
get_drive_file_list . . . . .	16
get_endpoint . . . . .	16
get_form_properties . . . . .	17
get_form_responses . . . . .	17
get_linked_quizzes_list . . . . .	18
get_materials_list . . . . .	18
get_materials_properties . . . . .	19
get_owner_id . . . . .	19
get_topic_list . . . . .	20
get_topic_properties . . . . .	20
is.google_forms_request . . . . .	21
make_form_quiz . . . . .	21
markdown_quiz_path . . . . .	22
ottr_quiz_to_google . . . . .	22
publish_coursework . . . . .	24
time_handler . . . . .	24
translate_questions_api . . . . .	25
update_form_settings . . . . .	25

<b>Index</b>	<b>27</b>
--------------	-----------

---

archive_course	<i>Archive a Google Classroom Course</i>
----------------	--

---

## Description

Archive a Google Classroom Course

## Usage

archive\_course(course\_id)

**Arguments**

course\_id      ID of the archived course you wish to delete

---

authorize      *Authorize R package to access Google classroom API*

---

**Description**

This is a function to authorize the R package to access the Googleclassroom API interactively.

**Usage**

```
authorize(token = NULL, cache = FALSE, ...)
```

**Arguments**

token            an output from `oauth2.0_token` to set as the authentication token.  
 cache            Should the token be cached as an `.httr-oauth` file?  
 ...              additional arguments to send to `oauth2.0_token`

**Value**

OAuth token saved to the environment so the package can use the users' Google data

**Examples**

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

---

auth\_from\_secret      *Use secrets to Authorize R package to access Google classroom API*

---

**Description**

This is a function to authorize the R package to access the Googleclassroom API. If no `client.id` and `client.secret` is provided, the package would provide predefined values.

**Usage**

```
auth_from_secret(access_token, refresh_token)
```

**Arguments**

access\_token    Access token can be obtained from running authorize interactively: token <- authorize(); token\$credentials\$access\_token

refresh\_token    Refresh token can be obtained from running authorize interactively: token <- authorize(); token\$credentials\$refresh\_token

**Value**

OAuth token saved to the environment so the package can use the users' Google data

**Examples**

```
## Not run:

token <- authorize()

auth_from_secret(
  token$credentials$access_token,
  token$credentials$refresh_token
)

## End(Not run)
```

---

commit_to_form	<i>Commit changes to a Google form</i>
----------------	--

---

**Description**

Commit changes to a Google form

**Usage**

```
commit_to_form(form_id, google_forms_request, quiet = FALSE)
```

**Arguments**

form\_id            The id of the google form to be updated

google\_forms\_request    The google slide request to be applied to the slides

quiet              TRUE/FALSE you'd like a progress message?

---

copy_form	<i>Make a copy of an existing form</i>
-----------	--

---

**Description**

Make a copy of an existing form

**Usage**

```
copy_form(form_id, new_name = NULL, quiet = FALSE)
```

**Arguments**

form_id	The form_id that is desired to be copied.
new_name	What should the new file name for the copied file be?
quiet	TRUE or FALSE whether messages should be printed out.

**Examples**

```
## Not run:  
#'  
# Make the form  
form_info <- copy_form(form_id = "https://docs.google.com/forms/d/someformidhere/edit",  
                        new_name = "copied form")  
  
## End(Not run)
```

---

create_course	<i>Create a new course</i>
---------------	----------------------------

---

**Description**

Create a new course

**Usage**

```
create_course(owner_id = get_owner_id())$id, name = NULL)
```

**Arguments**

owner_id	The ownerId to use to create the course. Will attempt to retrieve ownerId based on credentials with get_owner_id()
name	Name of the new course. Required.

**Examples**

```
## Not run:

owner_id <- get_owner_id()
course_df <- create_course(owner_id, name = "New course")

## End(Not run)
```

---

create\_coursework      *Create a new coursework*

---

**Description**

Create a new coursework

**Usage**

```
create_coursework(
  course_id = NULL,
  topic_id = NULL,
  publish = FALSE,
  title = NULL,
  work_type = "ASSIGNMENT",
  due_date = NULL,
  description = NULL,
  link = NULL
)
```

**Arguments**

course_id	Course id of where to make the new coursework. Can find from end of URL e.g. "https://classroom.google.com/c/COURSE_ID_IS_HERE"
topic_id	topic ID to be looked for.
publish	TRUE/FALSE, automatically publish the coursework upon posting? Default is to be posted as a draft (students will not see it until you click Post).
title	Name of new coursework. Required.
work_type	Currently only supported work type is ASSIGNMENT.
due_date	Required Due date for new coursework, must be given in year-month-day format.
description	Description of new coursework. Is a string
link	A url to an associated resource for the coursework being made.

## Examples

```
## Not run:
topic_id <- get_topic_list("604042323237")$topic$topicId[1]
course_id <- get_course_list()$courses$id[1]

create_coursework(course_id, topic_id,
  title = "a new quiz", due_date = "2025-12-1",
  description = "blah blah", link = "https://www.datatrail.org/"
)

## End(Not run)
```

---

create\_form

*Create a new form*

---

## Description

Create a new form

## Usage

```
create_form(title = NULL, document_title = "new_form", description = "")
```

## Arguments

**title**            The title for the new form. Required as a string.

**document\_title** The title for the form file that will be stored in Google Drive

**description**    The description for the new form as a string.

## Examples

```
## Not run:
#'
# Make the form
form_info <- create_form(
  title = "A great quiz",
  description = "This quiz is tricky"
)

## End(Not run)
```

---

create_material	<i>Create a new material</i>
-----------------	------------------------------

---

## Description

Create a new material

## Usage

```
create_material(  
  course_id = NULL,  
  topic_id = NULL,  
  publish = FALSE,  
  title = NULL,  
  description = NULL,  
  link = NULL  
)
```

## Arguments

course_id	Course id of where to make the new materials. Can find from end of URL e.g. "https://classroom.google.com/c/COURSE_ID_IS_HERE"
topic_id	topic ID to be looked for.
publish	TRUE/FALSE, automatically publish the coursework upon posting? Default is to be posted as a draft (students will not see it until you click Post).
title	Name of new material
description	A description for the new material
link	A URL to go with the associated material

## Examples

```
## Not run:  
course_id <- get_course_list()$courses$id[3]  
topic_id <- get_topic_list(course_id)$topic$topicId[1]  
  
create_material(course_id, topic_id, title = "new material")  
  
## End(Not run)
```



---

```
create_multiple_choice_question
    Create a multiple choice question
```

---

## Description

Create a multiple choice question

## Usage

```
create_multiple_choice_question(
    form_id = NULL,
    commit_to_form = TRUE,
    required = FALSE,
    question = NULL,
    choice_vector = NULL,
    shuffle_opt = FALSE,
    correct_answer = NULL,
    google_forms_request = NULL,
    point_value = 1,
    quiet = FALSE,
    location = 0
)
```

## Arguments

form_id	The id of the google form to be updated
commit_to_form	Whether or not the request should be committed. If need to build the request further, you will want to say FALSE. Default is TRUE
required	TRUE or FALSE is this a required question? Default is not required.
question	a string that is what the question should say
choice_vector	a character vector of the choices that should be given for this question
shuffle_opt	TRUE or FALSE options should be shuffled? default is FALSE
correct_answer	The index that corresponds to the correct answer in the 'choice_vector' supplied
google_forms_request	A google forms request object. If not supplied, it will be created new.
point_value	An integer representing how many points
quiet	TRUE/FALSE you'd like a progress message?
location	Where should the new question be added

**Examples**

```
## Not run:

create_multiple_choice_question(
  form_id = "12345",
  question = "What answer do you want?",
  choice_vector = c("A", "B", "C", "D"),
  correct_answer = 3,
  shuffle_opt = TRUE
)

## End(Not run)
```

---

```
create_quiz
```

```
Create a quiz at a course
```

---

**Description**

Create a quiz at a course

**Usage**

```
create_quiz(
  course_id = NULL,
  quiz_title = NULL,
  quiz_description = NULL,
  topic_id = NULL,
  coursework_title = "none",
  work_type = "ASSIGNMENT",
  due_date = NULL,
  assignment_description = ""
)
```

**Arguments**

<code>course_id</code>	A course id where the quiz should be created
<code>quiz_title</code>	A string indicating the title for the quiz
<code>quiz_description</code>	A description for the quiz that will be made
<code>topic_id</code>	Optional - a topic Id where the quiz will be posted
<code>coursework_title</code>	a string that will be what the coursework title
<code>work_type</code>	Currently only supported work type is ASSIGNMENT.
<code>due_date</code>	A due date for this quiz, in year-month-day format
<code>assignment_description</code>	The description that will be given for the assignment

**Examples**

```
## Not run:

course_id <- get_course_list()$courses$id[1]
topic_id <- get_topic_list(course_id)$topic$topicId[1]

create_quiz(course_id,
  quiz_title = "new quiz", quiz_description = "This is a great quiz",
  topic_id = topic_id, due_date = "2025-12-1"
)

## End(Not run)
```

---

```
create_text_question Create a text question
```

---

**Description**

This function makes a google request object that will be able to be posted with a batch request and added to a Google form to edit it.

**Usage**

```
create_text_question(
  form_id = NULL,
  commit_to_form = TRUE,
  required = FALSE,
  question = NULL,
  correct_answer = NULL,
  google_forms_request = NULL,
  point_value = 1,
  location = 0,
  quiet = FALSE
)
```

**Arguments**

form_id	The id of the google form to be updated
commit_to_form	Whether or not the request should be committed. If need to build the request further, you will want to say FALSE. Default is TRUE
required	TRUE or FALSE is this a required question? Default is not required.
question	a string that is what the question should say
correct_answer	A string that matches exactly what would be considered a correct
google_forms_request	A google forms request object. If not supplied, it will be created new.
point_value	An integer representing how many points
location	Where should the new question be added
quiet	TRUE/FALSE you'd like a progress message?

**Examples**

```
## Not run:

create_text_question(
  form_id = "12345",
  question = "Put text here that is for filling in the blank",
  point_value = 1
)

## End(Not run)
```

---

create_topic	<i>Create a new topic</i>
--------------	---------------------------

---

**Description**

Create a new topic

**Usage**

```
create_topic(course_id = NULL, name = NULL)
```

**Arguments**

course_id	Course id of where to make the new topic. Can find from end of URL e.g. "https://classroom.google.com/c/COURSE_ID_IS_HERE"
name	Name of new topic. Required.

---

date_handler	<i>Handle and parse a due_date</i>
--------------	------------------------------------

---

**Description**

Handle and parse a due\_date

**Usage**

```
date_handler(due_date)
```

**Arguments**

due_date	A string that is a date in format of year-month-day
----------	---

**Examples**

```
## Not run:
date_handler("2025-12-1")

## End(Not run)
```

---

delete_course	<i>Delete a Google Classroom Course</i>
---------------	---

---

**Description**

Delete a Google Classroom Course

**Usage**

```
delete_course(course_id)
```

**Arguments**

course_id	ID of the archived course you wish to delete
-----------	--

---

delete_coursework	<i>Delete a Google Classroom Coursework</i>
-------------------	---

---

**Description**

Delete a Google Classroom Coursework

**Usage**

```
delete_coursework(course_id, coursework_id)
```

**Arguments**

course_id	Course id of where to make the new coursework. Can find from end of URL e.g. "https://classroom.google.com/c/COURSE_ID_IS_HERE"
-----------	---

coursework_id	ID of the archived course you wish to delete
---------------	--

---

get\_coursework\_list    *Get list of courseworks for a course*

---

**Description**

Get list of courseworks for a course

**Usage**

```
get_coursework_list(course_id)
```

**Arguments**

course\_id    ID of the course to retrieve the courseworks from

**Examples**

```
## Not run:  
  
course_id <- get_course_list()$courses$id[1]  
  
get_coursework_list(course_id)  
  
## End(Not run)
```

---

get\_coursework\_properties  
                          *Get Google Classroom Course Properties*

---

**Description**

Get Google Classroom Course Properties

**Usage**

```
get_coursework_properties(course_id, coursework_id)
```

**Arguments**

course\_id    ID of the course you wish to retrieve information about a particular coursework  
coursework\_id    ID of the coursework you wish to retrieve information about

---

get\_course\_list      *Get list of courses*

---

**Description**

Get list of courses

**Usage**

```
get_course_list(owner_id = get_owner_id())$id
```

**Arguments**

owner\_id      owner\_id to retrieve course listings from

**Examples**

```
## Not run:  
  
owner_id <- get_owner_id()  
course_df <- get_course_list(owner_id)  
  
## End(Not run)
```

---

get\_course\_properties      *Get Google Classroom Course Properties*

---

**Description**

Get Google Classroom Course Properties

**Usage**

```
get_course_properties(course_id)
```

**Arguments**

course\_id      ID of the course you wish to retrieve information from

---

get_drive_file_list	<i>Get list of files from a Google Shared Drive</i>
---------------------	---

---

**Description**

Get list of files from a Google Shared Drive

**Usage**

```
get_drive_file_list(drive_id)
```

**Arguments**

drive_id	ID of the drive to retrieve a list of files from
----------	--

---

get_endpoint	<i>Generate endpoint for the Google classroom API</i>
--------------	---

---

**Description**

Generate endpoint for the Google classroom API

**Usage**

```
get_endpoint(
    type_of_endpoint = "classroom.endpoint.user",
    course_id = NULL,
    topic_id = NULL,
    coursework_id = NULL,
    materials_id = NULL,
    form_id = NULL
)
```

**Arguments**

type_of_endpoint	Type of endpoint to convert to url
course_id	(Optional) ID of the google course to be affected/retrieved
topic_id	(Optional) ID of the topic to be affected/retrieved
coursework_id	(Optional) ID of the coursework to be affected/retrieved
materials_id	(Optional) ID of the material to be affected/retrieved
form_id	(Optional) ID of the form to be affected/retrieved

**Value**

A url that is the endpoint for the API call



---

get\_form\_properties     *Get Google Form Properties*

---

**Description**

Get Google Form Properties

**Usage**

```
get_form_properties(form_id = NULL, form_url = NULL)
```

**Arguments**

form_id	Google form Id
form_url	Google form url

**Examples**

```
## Not run:  
  
# Make the form  
form_info <- create_form(title = "A great quiz", description = "This quiz is tricky")  
  
# Get info about the form  
form_info <- get_form_properties(form_id = form_info$formId)  
  
## End(Not run)
```

---

get\_form\_responses     *Get form responses*

---

**Description**

Get form responses

**Usage**

```
get_form_responses(form_id = NULL, form_url = NULL)
```

**Arguments**

form_id	The id of the google form to be updated
form_url	Google form url

**Examples**

```
## Not run:  
  
form_url <- "https://docs.google.com/forms/d/1woWtLVviIhrwb-NYEjVMO_IV2-62v0haS4N0/edit#responses"  
get_form_responses(form_url = form_url)  
  
## End(Not run)
```

---

```
get_linked_quizzes_list
```

*Get list of forms used in a course as quizzes*

---

**Description**

Get list of forms used in a course as quizzes

**Usage**

```
get_linked_quizzes_list(course_id)
```

**Arguments**

course\_id      ID of the course to retrieve the linked quizzes from

**Examples**

```
## Not run:  
course_id <- get_course_list()$courses$id[1]  
quiz_list <- get_linked_quizzes_list(course_id)  
  
## End(Not run)
```

---

```
get_materials_list
```

*Get list of materials for a course*

---

**Description**

Get list of materials for a course

**Usage**

```
get_materials_list(course_id)
```

**Arguments**

course\_id      ID of the course

### Examples

```
## Not run:  
course_id <- get_course_list()$courses$id[1]  
materials_df <- get_materials_list(course_id)  
  
## End(Not run)
```

---

get\_materials\_properties  
*Get Google Classroom Materials properties*

---

### Description

Get Google Classroom Materials properties

### Usage

```
get_materials_properties(course_id, materials_id)
```

### Arguments

course_id	ID of the course
materials_id	The material id you wish to retrieve

### Examples

```
## Not run:  
course_id <- get_course_list()$courses$id[3]  
materials_id <- get_materials_list(course_id)$material_id$courseWorkMaterial$id[1]  
  
get_materials_properties(course_id, materials_id)  
  
## End(Not run)
```

---

get\_owner\_id                    *Get ownerId based on credentials*

---

### Description

Get ownerId based on credentials

### Usage

```
get_owner_id()
```

---

get\_topic\_list      *Get list of topics for a course*

---

**Description**

Get list of topics for a course

**Usage**

```
get_topic_list(course_id)
```

**Arguments**

course\_id      ID of the course you wish to retrieve a topic list from

**Examples**

```
## Not run:  
  
course_id <- get_course_list()$courses$id[1]  
  
get_topic_list(course_id)  
  
## End(Not run)
```

---

get\_topic\_properties      *Get Google Classroom Topic Properties*

---

**Description**

Get Google Classroom Topic Properties

**Usage**

```
get_topic_properties(course_id, topic_id)
```

**Arguments**

course\_id      ID of the course  
topic\_id      topic ID to be looked for.

---

`is.google_forms_request`*Check if the object is a google forms request object*

---

**Description**

Check if the object is a google forms request object

**Usage**

```
is.google_forms_request(x)
```

**Arguments**

x                    A google\_forms\_request object created from the rgoogleclassroom package

---

`make_form_quiz`*Turn a form into a quiz*

---

**Description**

Turn a form into a quiz

**Usage**

```
make_form_quiz(form_id)
```

**Arguments**

form\_id             The id of the google form to be updated into a Quiz

**Examples**

```
## Not run:  
  
# Make the form  
form_info <- create_form(title = quiz_title)  
  
# Now make it a quiz  
make_form_quiz(form_id = form_info$formId)  
  
## End(Not run)
```

---

markdown\_quiz\_path     *Get file path to an example quiz*

---

**Description**

Get file path to an example quiz

**Usage**

```
markdown_quiz_path()
```

**Value**

A file path to a markua markdown quiz example you can use for testing

**Examples**

```
## Not run:  
  
# Find quiz path  
  
quiz_path <- markdown_quiz_path()  
  
## End(Not run)
```

---

ottr\_quiz\_to\_google     *Create Google Form Quiz from Markua quiz*

---

**Description**

Takes a file path to a Markua formatted quiz and runs the steps to convert it to a Google Form Request and sends It to be a Google form quiz.

**Usage**

```
ottr_quiz_to_google(  
  quiz_path = NULL,  
  course_id = NULL,  
  quiz_title = NULL,  
  topic_id = NULL,  
  coursework_title = NULL,  
  form_id = NULL,  
  due_date = NULL,  
  make_new_quiz = FALSE,  
  copy_from_template_quiz = TRUE,
```

```

    new_name = NULL,
    assignment_description = "",
    quiz_description = "",
    output_path = NULL,
    quiet = FALSE
  )

```

### Arguments

quiz_path	file path to a markdown Markua quiz
course_id	An id for the course where this is to be published and linked.
quiz_title	The title for the quiz. If not supplied, it will attempt to be grabbed from the Markua doc
topic_id	topic ID that the quiz should be added under.
coursework_title	the title for the coursework to be created
form_id	form id where this quiz is to be published. Alternatively, if you want a new quiz to be made, you should set make_new_quiz = TRUE and leave this NULL.
due_date	A due date for this quiz, in year-month-day format
make_new_quiz	This can only be used if form_id is not specified. This will make a new quiz
copy_from_template_quiz	TRUE or FALSE the form supplied should be copied over and used as a template.
new_name	To be passed to 'copy_form' if 'copy_from_template_quiz' is TRUE. What the new file name should be called
assignment_description	The description that will be given for the assignment
quiz_description	The description that will be given for the quiz
output_path	Optional file path to save the question formatted data to
quiet	TRUE/FALSE you'd like a progress message?

### Examples

```

## Not run:

# Using quiz example

quiz_path <- markdown_quiz_path()

ottr_quiz_to_google(
  markdown_quiz_path(),
  course_id = "606463350924",
  make_new_quiz = TRUE,
  due_date = "2025-12-1"
)

```

```
## End(Not run)
```

---

```
publish_coursework      Publish a Google Classroom CourseWork
```

---

**Description**

Publish a Google Classroom CourseWork

**Usage**

```
publish_coursework(course_id, coursework_id)
```

**Arguments**

```
course_id      ID of the archived course you wish to delete
coursework_id  coursework ID of the coursework you wish to publish
```

---

```
time_handler      Handle and parse a time
```

---

**Description**

Handle and parse a time

**Usage**

```
time_handler(due_time = lubridate::hms("23:59:59"))
```

**Arguments**

```
due_time      A string that is a date in format of year-month-day. Default is midnight.
```

**Examples**

```
## Not run:
time_handler("21:30:59")

## End(Not run)
```



---

`translate_questions_api`*Translate Markua questions for submission to Google API*

---

**Description**

Takes a Markua formatted quiz and translates it to something that can be sent to Google Forms API.

**Usage**

```
translate_questions_api(quiz_path, output_path = NULL)
```

**Arguments**

<code>quiz_path</code>	The file path to a Markua formatted quiz to be translated to Google Forms API format
<code>output_path</code>	Optional file path to save the formatted data to a JSON file

**Value**

A list of the output from [ottrpal::check\_question] with messages/warnings regarding each question and each check.

**Examples**

```
## Not run:  
  
# Using quiz example  
  
quiz_path <- markdown_quiz_path()  
parsed_questions <- translate_questions_api(quiz_path)  
  
## End(Not run)
```

---

`update_form_settings` *Create a multiple choice question*

---

**Description**

Create a multiple choice question

**Usage**

```
update_form_settings(  
    form_id = NULL,  
    title = NULL,  
    description = NULL,  
    google_forms_request = NULL,  
    quiet = FALSE  
)
```

**Arguments**

form_id	The id of the google form to be updated
title	An updated title
description	An updated description
google_forms_request	A google forms request object. If not supplied, it will be created new.
quiet	TRUE/FALSE you'd like a progress message?

**Examples**

```
## Not run:  
  
update_form_settings(  
    form_id = "12345",  
    new_title = NULL,  
    new_description = NULL  
)  
  
## End(Not run)
```

# Index

[archive\\_course](#), 2  
[auth\\_from\\_secret](#), 3  
[authorize](#), 3

[commit\\_to\\_form](#), 4  
[copy\\_form](#), 5  
[create\\_course](#), 5  
[create\\_coursework](#), 6  
[create\\_form](#), 7  
[create\\_material](#), 8  
[create\\_multiple\\_choice\\_question](#), 9  
[create\\_quiz](#), 10  
[create\\_text\\_question](#), 11  
[create\\_topic](#), 12

[date\\_handler](#), 12  
[delete\\_course](#), 13  
[delete\\_coursework](#), 13

[get\\_course\\_list](#), 15  
[get\\_course\\_properties](#), 15  
[get\\_coursework\\_list](#), 14  
[get\\_coursework\\_properties](#), 14  
[get\\_drive\\_file\\_list](#), 16  
[get\\_endpoint](#), 16  
[get\\_form\\_properties](#), 17  
[get\\_form\\_responses](#), 17  
[get\\_linked\\_quizzes\\_list](#), 18  
[get\\_materials\\_list](#), 18  
[get\\_materials\\_properties](#), 19  
[get\\_owner\\_id](#), 19  
[get\\_topic\\_list](#), 20  
[get\\_topic\\_properties](#), 20

[is.google\\_forms\\_request](#), 21

[make\\_form\\_quiz](#), 21  
[markdown\\_quiz\\_path](#), 22

[oauth2.0\\_token](#), 3  
[ottr\\_quiz\\_to\\_google](#), 22

[publish\\_coursework](#), 24

[time\\_handler](#), 24  
[translate\\_questions\\_api](#), 25

[update\\_form\\_settings](#), 25