

Package: canvasquiz (via r-universe)

July 5, 2026

Title Quiz For Canvas Learning Management System

Version 0.0.9000

Description Create, manipulate and get classic quizzes directly via
Canvas learning management system's API.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Imports cli, commonmark, curl, dplyr, glue, httr2, tibble, tidyr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

URL <https://github.com/emitanaka/canvasquiz>,
<http://emitanaka.org/canvasquiz/>

BugReports <https://github.com/emitanaka/canvasquiz/issues>

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2026-04-21 22:42:45 UTC

RemoteUrl <https://github.com/emitanaka/canvasquiz>

RemoteRef HEAD

RemoteSha 4c3521de267acad2a07a313fbd02f073fe61411b

Contents

add_question_from_bank	3
answer_essay	4
answer_fill_in_multiple_blanks	4

answer_matching	4
answer_mcq	5
answer_multiple	5
answer_none	6
answer_num	6
answer_single	7
answer_text	8
answer_true_false	9
answer_upload_file	9
attempt-options	10
attempt_options	10
canvas	11
count_submissions	12
create_question	13
create_quiz	14
delete_all_quizzes	15
delete_quiz	16
delete_quiz_question	16
dropdown	17
fill_in_the_blank	17
last-ids	18
list_attempted_questions	18
list_files	19
list_folder	19
list_pages	20
list_questions	20
list_quizzes	21
list_submissions	22
md	23
quiz_questions	23
quiz_results	24
quiz_results_summary	24
quiz_statistics	25
quiz_submissions	26
regrade_question	26
submission_info	27
submission_overview	28
submission_questions	29
submission_update	30
tag_file	30
tag_img	31
update_question	32
upload_file	33
Index	34

`add_question_from_bank`*Add questions from a question bank to a quiz*

Description

Add questions from a question bank to a quiz

Usage

```
add_question_from_bank(  
    quiz_id,  
    bank_id,  
    title = NULL,  
    n = 1L,  
    points = 1L,  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

<code>quiz_id</code>	The id of the quiz to add questions to.
<code>bank_id</code>	The id of the question bank to pull questions from.
<code>title</code>	(Optional) The title of the quiz group to create for these questions. If NULL (default), no quiz group will be created and questions will be added to the quiz without a group.
<code>n</code>	The number of questions to randomly select from the question bank. Defaults to 1.
<code>points</code>	The number of points each question added from the bank should be worth. Defaults to 1.
<code>course_id</code>	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
<code>url</code>	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
<code>token</code>	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

The API response from adding the questions.

answer_essay *Answer for an essay question*

Description

Answer for an essay question

Usage

answer_essay()

See Also

Other answer-functions: [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_fill_in_multiple_blanks
Answer for a fill-in-multiple-blanks question

Description

Answer for a fill-in-multiple-blanks question

Usage

answer_fill_in_multiple_blanks(answers, blank_ids)

answer_matching *Answer for a matching question*

Description

Answer for a matching question

Usage

answer_matching(left, right, extra_choices = "")

Arguments

left	Character vector of the left-hand side items to be matched.
right	Character vector of the right-hand side items to be matched. Must be the same length as left.
extra_choices	Character vector of extra choices that can be used as incorrect matches. Optional.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_mcq	<i>Answer for a multiple choice question</i>
------------	--

Description

Answer for a multiple choice question

Usage

```
answer_mcq(correct, choices, multiple = length(correct) > 1)
```

Arguments

correct	Character vector of correct answer(s). Should be one element for single-answer multiple choice questions and can be multiple elements for multiple-answer multiple choice questions.
choices	Character vector of answer choices.
multiple	Logical. Whether the question allows multiple correct answers. Defaults to TRUE if correct has more than one element.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_multiple	<i>Multiple answer question</i>
-----------------	---------------------------------

Description

This function allows you to create a question with multiple parts, such as multiple dropdowns or fill-in-the-blank questions with multiple blanks. Each part is created with a separate answer object (e.g. created by [dropdown\(\)](#) or [fill_in_the_blank\(\)](#)) and then combined into a single question with this function.

Usage

```
answer_multiple(...)
```

Arguments

...	Multiple dropdown answer objects created by dropdown() or fill_in_the_blank() ..
-----	--

Details

The question text should include square-bracketed text corresponding to the id of each answer part to indicate where in the question text each part should be displayed. For example, if you have two dropdown answer parts with ids "blank1" and "blank2", your question text might look like "The capital of France is [blank1] and the capital of Spain is [blank2]."

All answer parts must be of the same type (e.g. all dropdowns or all fill-in-the-blank). The function will throw an error if you try to combine different types of answer parts or if the question text includes blank ids that do not correspond to any answer part ids.

For fill-in-the-blank questions, the answer may be marked incorrect even if it is technically correct. This seems to be the result of encoding. If you manually edit and resave the questions, this seems to fix the issue.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_none	<i>A text-only question with no answers</i>
-------------	---

Description

A text-only question with no answers

Usage

```
answer_none()
```

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_num	<i>A numerical answer question with an exact answer</i>
------------	---

Description

A numerical answer question with an exact answer

Usage

```
answer_num(value, tol = 0)
```

```
answer_num_precision(value, precision = 0L)
```

```
answer_num_range(lower, upper = lower)
```

Arguments

value	The correct numerical answer.
tol	The acceptable error margin for the answer. Defaults to 0 for an exact answer.
precision	The number of decimal places the answer must be correct to.
lower	The lower bound of the acceptable answer range.
upper	The upper bound of the acceptable answer range.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_single	<i>Create an answer object for a quiz question</i>
---------------	--

Description

This function is a wrapper that creates the appropriate answer object based on the input. It can handle multiple choice questions, true/false questions, short answer questions, numerical questions (with exact answers, precision-based answers, or range-based answers), and matching questions. The function will automatically determine the type of question based on the format of the input and create the corresponding answer object using the appropriate helper functions.

Usage

```
answer_single(x, y = NULL, tol = 0, precision = 0)
```

Arguments

x	The correct answer(s) for the question. See details for the expected format based on the type of question.
y	Additional parameter for certain question types. See details for the expected format based on the type of question.
tol	The acceptable error margin for the answer. Defaults to 0 for an exact answer.
precision	The number of decimal places the answer must be correct to.

Details

The function uses the following logic to determine the type of question:

- If x is a logical value and y is NULL, it creates a true/false question using [answer_true_false\(\)](#).
- If x is a single numeric value and y is NULL, it creates a numerical question with an exact answer using [answer_num\(\)](#). If tol is greater than 0, it creates a numerical question with an error margin. If precision is greater than 0, it creates a numerical question with a specified precision.

- If `x` is a numeric vector of length 2, it creates a numerical question with a range-based answer using `answer_num_range()`. Alternatively, if `x` is a single numeric value and `y` is a single numeric value, it also creates a numerical question with a range-based answer using `answer_num_range()`, where `x` is the lower bound and `y` is the upper bound.
- If `x` is a named character vector, it creates a matching question using `answer_matching()`. Any extra choices for the matching question can be provided in `y`.
- If `x` and `y` are character vectors, it creates a multiple choice question using `answer_mcq()`.
- If the input format does not match any of the above cases, the function throws an error indicating that the answer format is unsupported.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_text

A short answer question

Description

A short answer question

Usage

```
answer_text(correct)
```

Arguments

`correct` The correct answer text.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_true_false\(\)](#), [answer_upload_file\(\)](#)

answer_true_false *True or False Question*

Description

True or False Question

Usage

```
answer_true_false(correct = TRUE)
```

Arguments

correct Logical. Whether the correct answer is TRUE or FALSE.

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_upload_file\(\)](#)

answer_upload_file *A file upload question*

Description

A file upload question

Usage

```
answer_upload_file()
```

See Also

Other answer-functions: [answer_essay\(\)](#), [answer_matching\(\)](#), [answer_mcq\(\)](#), [answer_multiple\(\)](#), [answer_none\(\)](#), [answer_num\(\)](#), [answer_single\(\)](#), [answer_text\(\)](#), [answer_true_false\(\)](#)

attempt-options	<i>Options for showing quiz results or answers</i>
-----------------	--

Description

Options for showing quiz results or answers

Usage

```
opt_show(last_attempt = FALSE, one_time = FALSE, at = NULL)
```

```
opt_hide(at = NULL)
```

```
opt_shuffle()
```

Arguments

<code>last_attempt</code>	logical. Whether to show results/answers only after the last attempt is submitted. Only valid if the number of attempts is greater than 1. Defaults to FALSE.
<code>one_time</code>	logical. Whether to show results/answers only immediately after submission.
<code>at</code>	POSIXct. Date and time to control when results/answers become visible or hidden. If <code>show = TRUE</code> , results/answers become visible at this date and time. If <code>hide = TRUE</code> , results/answers become hidden at this date/time.

Value

A list of options to be passed to the `results` or `answer` parameters of `attempt_options()`.

attempt_options	<i>A quiz attempt options</i>
-----------------	-------------------------------

Description

A quiz attempt options

Usage

```
attempt_options(
  n = 1,
  time = NULL,
  due = NULL,
  lock = NULL,
  unlock = NULL,
  results = opt_show(at = due),
  answer = opt_show(at = due),
```

```

    score = c("highest", "latest"),
    access_code = NULL,
    ip_filter = NULL
)

```

Arguments

<code>n</code>	The number of allowed attempts. Set to -1 or Inf for unlimited. Defaults to 1.
<code>time</code>	Time limit for each attempt in minutes. Set to NULL for no limit (default).
<code>due</code>	POSIXct. Due date and time for the quiz.
<code>lock</code>	POSIXct. Lock date and time for the quiz.
<code>unlock</code>	POSIXct. Unlock date and time for the quiz.
<code>results</code>	A list of options controlling when quiz results are visible to students. See opt_show() and opt_hide() .
<code>answer</code>	A list of options controlling when correct answers are visible to students. See opt_show() , opt_hide() and opt_shuffle() .
<code>score</code>	Scoring policy for multiple attempts. Allowed values: "highest" (default), "latest".
<code>access_code</code>	Password required to access the quiz. Set to NULL for no restriction (default).
<code>ip_filter</code>	Restrict access to computers in a specified IP range. Filters can be a comma-separated list of addresses, or address/mask. Examples: "192.168.1.1,192.168.1.2" or "192.168.1.0/24". Set to NULL for no restriction (default).

Value

A list of options to be passed to [create_quiz\(\)](#).

<code>canvas</code>	<i>Set and get Canvas course, URL, and token</i>
---------------------	--

Description

These functions set and retrieve the Canvas course ID, URL, and token from environment variables. This allows for easy configuration of API calls to the Canvas API without hardcoding sensitive information in the code.

Usage

```

set_canvas_course_id(course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"))

get_canvas_course_id()

set_canvas_url(url = Sys.getenv("CANVASQUIZ_URL"))

get_canvas_url()

```

```
set_canvas_token(token = Sys.getenv("CANVASQUIZ_TOKEN"))

get_canvas_token()
```

Arguments

course_id	The Canvas course ID to use for API calls.
url	The base URL for the Canvas instance (e.g., "https:// canvas.instructure.com").
token	The API token for authenticating with the Canvas API.

count_submissions	<i>Count the number of quiz submissions</i>
-------------------	---

Description

Count the number of quiz submissions

Usage

```
count_submissions(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

quiz_id	The ID of the quiz to count submissions for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

create_question	<i>Create a Quiz Question</i>
-----------------	-------------------------------

Description

This function creates a quiz question directly into Canvas. You can create a variety of question types, including multiple choice, short answer, numerical answer, matching, essay, file upload, and text-only questions.

Usage

```
create_question(
  text = NULL,
  answers = NULL,
  points = 1,
  quiz_id = last_quiz_id(),
  quiz_group_id = NULL,
  position = NULL,
  title = NULL,
  correct_comments = NULL,
  incorrect_comments = NULL,
  neutral_comments = NULL,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

text	Character. The text of the question.
answers	A list of answer objects.
points	The maximum amount of points received for answering this question correctly.
quiz_id	The id of the quiz to add the question to.
quiz_group_id	The id of the quiz group to add the question to.
position	An integer specifying the order in which the question will be displayed in the quiz. Doesn't seem to work.
title	The name of the question.
correct_comments	Comment to display if the student answers the question correctly.
incorrect_comments	Comment to display if the student answers incorrectly.
neutral_comments	Comment to display regardless of how the student answered.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.

url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Details

The question comments is correctly entered, however, Canvas doesn't seem to display them initially. Manually editing the quiz and saving it again seems to fix the issue and then the comments are displayed as expected.

Value

A list representing the quiz question.

create_quiz	<i>Create a Quiz</i>
-------------	----------------------

Description

Create a quiz with a number of customizable parameters.

Usage

```
create_quiz(
    title,
    description = NULL,
    quiz_type = c("practice_quiz", "assignment", "graded_survey", "survey"),
    one_question_at_a_time = c("no", "yes", "yes_but_cant_go_back"),
    attempts = attempt_options(),
    publish = FALSE,
    visible = c("everyone", "none"),
    assignment_group_id = NULL,
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
    url = Sys.getenv("CANVASQUIZ_URL"),
    token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

title	The quiz title (required).
description	A description of the quiz.
quiz_type	The type of quiz. Allowed values: "practice_quiz", "assignment", "graded_survey", "survey".
one_question_at_a_time	Whether to show one question at a time. Allowed values: "no" (default), "yes", "yes_but_cant_go_back".

attempts	A list of options controlling quiz attempts, as created by the <code>attempt_options()</code> function.
publish	Whether to publish the quiz immediately. Defaults to FALSE.
visible	Who this quiz is visible to. Allowed values: "everyone" (default) or "none".
assignment_group_id	integer. The assignment group ID to put the assignment in. Defaults to the top assignment group in the course. Only valid if quiz_type is "assignment" or "graded_survey".
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

The quiz id.

delete_all_quizzes *Delete all quizzes*

Description

Use with caution! This function will delete all quizzes in the course after asking for confirmation. It will not delete quiz question banks.

Usage

```
delete_all_quizzes(
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

delete_quiz *Delete a quiz*

Description

Delete a quiz

Usage

```
delete_quiz(  
    quiz_id,  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

quiz_id	The id of the quiz to retrieve questions from.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

delete_quiz_question *Delete a quiz question*

Description

Delete a quiz question

Usage

```
delete_quiz_question(  
    quiz_id,  
    question_id,  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

quiz_id	The id of the quiz to retrieve questions from.
question_id	The id of the question to delete.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

dropdown	<i>Create a dropdown answer for a fill-in-multiple-blanks question</i>
----------	--

Description

Create a dropdown answer for a fill-in-multiple-blanks question

Usage

```
dropdown(correct, choices, id = NULL)
```

Arguments

correct	Character vector of correct answer(s). Should be one element for single-answer multiple choice questions and can be multiple elements for multiple-answer multiple choice questions.
choices	Character vector of answer choices.
blank_id	The ID of the blank this dropdown corresponds to.

fill_in_the_blank	<i>Create a fill-in-the-blank answer for a fill-in-multiple-blanks question</i>
-------------------	---

Description

Create a fill-in-the-blank answer for a fill-in-multiple-blanks question

Usage

```
fill_in_the_blank(correct, id = NULL)
```

Arguments

correct	The correct answer text for this blank.
id	The ID of the blank this answer corresponds to.

last-ids	<i>Get the ID of the most recently created quiz in a course</i>
----------	---

Description

Get the ID of the most recently created quiz in a course

Usage

```
last_quiz_id()
```

```
last_question_id()
```

Value

The ID of the most recently created quiz in the specified course.

list_attempted_questions	<i>List attempted questions for a quiz</i>
--------------------------	--

Description

List attempted questions for a quiz

Usage

```
list_attempted_questions(  
    quiz_id,  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

See Also

Other submissions: [list_submissions\(\)](#), [submission_info\(\)](#), [submission_overview\(\)](#), [submission_questions\(\)](#)

list_files	<i>List files in a folder or course</i>
------------	---

Description

List files in a folder or course

Usage

```
list_files(  
    folder_id = NULL,  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

folder_id	The ID of the folder to list files from. If NULL, lists files in the course.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

list_folder	<i>List the folder in a course</i>
-------------	------------------------------------

Description

List the folder in a course

Usage

```
list_folder(  
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),  
    url = Sys.getenv("CANVASQUIZ_URL"),  
    token = Sys.getenv("CANVASQUIZ_TOKEN"),  
    tz = Sys.timezone()  
)
```

Arguments

course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

list_pages	<i>List pages in a course</i>
------------	-------------------------------

Description

List pages in a course

Usage

```
list_pages(
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

list_questions	<i>List quiz questions in a course</i>
----------------	--

Description

List quiz questions in a course

Usage

```
list_questions(
  quiz_id = NULL,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)

list_question_groups(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

quiz_id	The id of the quiz to retrieve questions from.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

A data frame of quiz questions with their details.

list_quizzes	<i>List quizzes in a course</i>
--------------	---------------------------------

Description

Currently only a maximum of 100 quizzes will be returned.

Usage

```
list_quizzes(
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

A data frame of quizzes with their details.

list_submissions	<i>List quiz submissions</i>
------------------	------------------------------

Description

List quiz submissions

Usage

```
list_submissions(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN"),
  n = count_submissions(quiz_id, course_id, url, token),
  tz = Sys.timezone()
)
```

Arguments

quiz_id	The ID of the quiz to retrieve statistics for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.
n	The maximum number of quiz submissions to retrieve.
tz	The timezone to use for the started_at and finished_at columns. Defaults to the system timezone.

See Also

Other submissions: [list_attempted_questions\(\)](#), [submission_info\(\)](#), [submission_overview\(\)](#), [submission_questions\(\)](#)

md	<i>Convert Markdown to HTML</i>
----	---------------------------------

Description

Convert Markdown to HTML

Usage

```
md(text)
```

Arguments

text	A character string containing Markdown text.
------	--

quiz_questions	<i>Get quiz questions</i>
----------------	---------------------------

Description

Get quiz questions

Usage

```
quiz_questions(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

quiz_id	The id of the quiz to retrieve questions from.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

A list of quiz questions.

quiz_results	<i>Quiz results with user information</i>
--------------	---

Description

Quiz results with user information

Usage

```
quiz_results(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN"),
  n = count_submissions(quiz_id, course_id, url, token),
  tz = Sys.timezone()
)
```

Arguments

quiz_id	The ID of the quiz to retrieve statistics for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.
n	The maximum number of quiz submissions to retrieve.
tz	The timezone to use for the started_at and finished_at columns. Defaults to the system timezone.

quiz_results_summary	<i>Quiz results summary</i>
----------------------	-----------------------------

Description

Quiz results summary

Usage

```

quiz_results_summary(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN"),
  n = count_submissions(quiz_id, course_id, url, token),
  tz = Sys.timezone()
)

```

Arguments

quiz_id	The ID of the quiz to retrieve statistics for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.
n	The maximum number of quiz submissions to retrieve.
tz	The timezone to use for the started_at and finished_at columns. Defaults to the system timezone.

quiz_statistics	<i>Get quiz statistics</i>
-----------------	----------------------------

Description

Get quiz statistics

Usage

```

quiz_statistics(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)

```

Arguments

quiz_id	The ID of the quiz to retrieve statistics for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.

url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

quiz_submissions	<i>Get quiz submissions with question-level answers</i>
------------------	---

Description

Get quiz submissions with question-level answers

Usage

```
quiz_submissions(
  quiz_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN"),
  n = count_submissions(quiz_id, course_id, url, token)
)
```

Arguments

quiz_id	The ID of the quiz to retrieve statistics for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.
n	The maximum number of quiz submissions to retrieve.

regrade_question	<i>Regrade quiz submissions based on answer</i>
------------------	---

Description

If you leave the answer as NULL, it will not filter the submissions.

Usage

```

regrade_question(
    question_id,
    quiz_id,
    answer = NULL,
    fudge_points = NULL,
    score = NULL,
    comment = NULL,
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
    url = Sys.getenv("CANVASQUIZ_URL"),
    token = Sys.getenv("CANVASQUIZ_TOKEN")
)

```

Arguments

question_id	The ID of the question to regrade.
answer	The answer to regrade. Can be numeric or character or one of the answer functions.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

submission_info	<i>Get quiz submission details</i>
-----------------	------------------------------------

Description

Get quiz submission details

Usage

```

submission_info(
    submission_id,
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
    url = Sys.getenv("CANVASQUIZ_URL"),
    token = Sys.getenv("CANVASQUIZ_TOKEN")
)

```

Arguments

submission_id	The ID of the quiz submission to retrieve details for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

A data frame with submission id, quiz id, question name, question type, question text, whether the question was flagged, whether the question was correct, assessment question id, quiz group id, and the answers provided.

See Also

Other submissions: [list_attempted_questions\(\)](#), [list_submissions\(\)](#), [submission_overview\(\)](#), [submission_questions\(\)](#)

submission_overview *Get quiz submission overview*

Description

Get quiz submission overview

Usage

```
submission_overview(
  submission_id,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

submission_id	The ID of the quiz submission to retrieve details for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Value

A data frame with the name of the user, quiz title, score, attempt number, quiz id, and submission id.

See Also

Other submissions: [list_attempted_questions\(\)](#), [list_submissions\(\)](#), [submission_info\(\)](#), [submission_questions\(\)](#)

submission_questions *List attempted questions for a quiz*

Description

List attempted questions for a quiz

Usage

```
submission_questions(  
  submission_id,  
  url = Sys.getenv("CANVASQUIZ_URL"),  
  token = Sys.getenv("CANVASQUIZ_TOKEN")  
)
```

Arguments

submission_id	The ID of the quiz submission to retrieve details for.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

See Also

Other submissions: [list_attempted_questions\(\)](#), [list_submissions\(\)](#), [submission_info\(\)](#), [submission_overview\(\)](#)

submission_update	<i>Update a quiz submission</i>
-------------------	---------------------------------

Description

Update a quiz submission

Usage

```
submission_update(
    submission_id,
    fudge_points = NULL,
    question_id = NULL,
    score = NULL,
    comment = NULL,
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
    url = Sys.getenv("CANVASQUIZ_URL"),
    token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

submission_id	The ID of the quiz submission to retrieve details for.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

tag_file	<i>Generate an HTML link to download a file in a Canvas course This function creates an HTML link that allows users to download a file stored in a Canvas course. The link will point to the download URL for the specified file.</i>
----------	---

Description

Generate an HTML link to download a file in a Canvas course This function creates an HTML link that allows users to download a file stored in a Canvas course. The link will point to the download URL for the specified file.

Usage

```

tag_file(file_id, text = "Download")

upload_tag_file(
  file_path,
  text = "Download",
  folder_id = NULL,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)

```

Arguments

`file_id` The file ID of the file in Canvas.

`text` The text to display for the download link (default is "Download").

Value

A character string containing the HTML link to download the specified file.

tag_img	<i>Generate an HTML image tag for a file in a Canvas course</i>
---------	---

Description

This function creates an HTML `` tag that can be used to display an image file stored in a Canvas course. The image will be displayed at 50% width.

Usage

```

tag_img(
  file_id,
  width = "100%",
  class = "",
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID")
)

upload_tag_img(
  file_path,
  width = "100%",
  class = "",
  folder_id = NULL,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)

```

Arguments

file_id	The file ID of the image in Canvas.
width	The width of the image (default is "100%").
class	Optional CSS class to apply to the tag.
course_id	The ID of the Canvas course where the file is stored (default is taken from the environment variable CANVASQUIZ_COURSE_ID).

Value

A character string containing the HTML tag for the specified image file.

update_question	<i>Update a quiz question</i>
-----------------	-------------------------------

Description

Update a quiz question

Usage

```
update_question(
  question_id,
  quiz_id,
  text = NULL,
  answers = NULL,
  points = NULL,
  position = NULL,
  title = NULL,
  correct_comments = NULL,
  incorrect_comments = NULL,
  neutral_comments = NULL,
  course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
  url = Sys.getenv("CANVASQUIZ_URL"),
  token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

quiz_id	The id of the quiz to add the question to.
text	Character. The text of the question.
answers	A list of answer objects.
points	The maximum amount of points received for answering this question correctly.
position	An integer specifying the order in which the question will be displayed in the quiz. Doesn't seem to work.

title	The name of the question.
correct_comments	Comment to display if the student answers the question correctly.
incorrect_comments	Comment to display if the student answers incorrectly.
neutral_comments	Comment to display regardless of how the student answered.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

upload_file	<i>Upload a file to a folder</i>
-------------	----------------------------------

Description

Upload a file to a folder

Usage

```
upload_file(
    file,
    folder_id = NULL,
    course_id = Sys.getenv("CANVASQUIZ_COURSE_ID"),
    url = Sys.getenv("CANVASQUIZ_URL"),
    token = Sys.getenv("CANVASQUIZ_TOKEN")
)
```

Arguments

file	The path to the file to upload.
folder_id	The ID of the folder to upload the file to.
course_id	The course id. Defaults to the value of the CANVASQUIZ_COURSE_ID environment variable.
url	The canvas url. Defaults to the value of the CANVASQUIZ_URL environment variable.
token	The canvas token. Defaults to the value of the CANVASQUIZ_TOKEN environment variable.

Index

- * **answer-functions**
 - answer_essay, 4
 - answer_matching, 4
 - answer_mcq, 5
 - answer_multiple, 5
 - answer_none, 6
 - answer_num, 6
 - answer_single, 7
 - answer_text, 8
 - answer_true_false, 9
 - answer_upload_file, 9
- * **submissions**
 - list_attempted_questions, 18
 - list_submissions, 22
 - submission_info, 27
 - submission_overview, 28
 - submission_questions, 29
- add_question_from_bank, 3
- answer_essay, 4, 5–9
- answer_fill_in_multiple_blanks, 4
- answer_matching, 4, 4–9
- answer_mcq, 4, 5, 5–9
- answer_multiple, 4, 5, 5–9
- answer_none, 4, 5, 6, 6–9
- answer_num, 4, 5, 6, 6, 8, 9
- answer_num_precision (answer_num), 6
- answer_num_range (answer_num), 6
- answer_single, 4–6, 7, 7–9
- answer_text, 4–7, 8, 8, 9
- answer_true_false, 4–8, 9, 9
- answer_upload_file, 4–8, 9, 9
- attempt-options, 10
- attempt_options, 10
- attempt_options(), 10, 15
- canvas, 11
- count_submissions, 12
- create_question, 13
- create_quiz, 14
- create_quiz(), 11
- delete_all_quizzes, 15
- delete_quiz, 16
- delete_quiz_question, 16
- dropdown, 17
- dropdown(), 5
- fill_in_the_blank, 17
- fill_in_the_blank(), 5
- get_canvas_course_id (canvas), 11
- get_canvas_token (canvas), 11
- get_canvas_url (canvas), 11
- last-ids, 18
- last_question_id (last-ids), 18
- last_quiz_id (last-ids), 18
- list_attempted_questions, 18, 22, 28, 29
- list_files, 19
- list_folder, 19
- list_pages, 20
- list_question_groups (list_questions), 20
- list_questions, 20
- list_quizzes, 21
- list_submissions, 18, 22, 28, 29
- md, 23
- opt_hide (attempt-options), 10
- opt_hide(), 11
- opt_show (attempt-options), 10
- opt_show(), 11
- opt_shuffle (attempt-options), 10
- opt_shuffle(), 11
- quiz_questions, 23
- quiz_results, 24
- quiz_results_summary, 24
- quiz_statistics, 25

quiz_submissions, [26](#)

regrade_question, [26](#)

set_canvas_course_id (canvas), [11](#)

set_canvas_token (canvas), [11](#)

set_canvas_url (canvas), [11](#)

submission_info, [18](#), [22](#), [27](#), [29](#)

submission_overview, [18](#), [22](#), [28](#), [28](#), [29](#)

submission_questions, [18](#), [22](#), [28](#), [29](#), [29](#)

submission_update, [30](#)

tag_file, [30](#)

tag_img, [31](#)

update_question, [32](#)

upload_file, [33](#)

upload_tag_file (tag_file), [30](#)

upload_tag_img (tag_img), [31](#)