

# Package ‘vvcanvas’

April 29, 2024

**Title** 'Canvas' LMS API Integration

**Version** 0.0.4

**Description**

Allow R users to interact with the 'Canvas' Learning Management System (LMS) API (see [https://canvas.instructure.com/doc/api/all\\_resources.html](https://canvas.instructure.com/doc/api/all_resources.html) for details). It provides a set of functions to access and manipulate course data, assignments, grades, users, and other resources available through the 'Canvas' API.

**URL** <https://github.com/vusaverse/vvcanvas>,  
<https://vusaverse.github.io/vvcanvas/>

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** dplyr, htm2txt, httr, jsonlite, magrittr, purrr, rlang,  
stringr, tidyr, utils

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Tomer Iwan [aut, cre, cph]

**Maintainer** Tomer Iwan <t.iwan@vu.nl>

**Repository** CRAN

**Date/Publication** 2024-04-29 09:10:04 UTC

## R topics documented:

canvas_api_key . . . . .	3
canvas_authenticate . . . . .	3
canvas_base_url . . . . .	4
create_assignment_group . . . . .	4
create_course_datalake . . . . .	5
create_course_section . . . . .	5

create_folder . . . . .	6
create_group_category . . . . .	7
create_page . . . . .	7
delete_course_section . . . . .	8
download_course_file . . . . .	8
edit_section . . . . .	9
get_accounts . . . . .	10
get_all_courses . . . . .	11
get_assignments . . . . .	11
get_assignment_data . . . . .	12
get_assignment_details . . . . .	12
get_assignment_submissions . . . . .	13
get_calendar_events . . . . .	13
get_courses . . . . .	15
get_course_announcements . . . . .	15
get_course_details . . . . .	16
get_course_enrollments . . . . .	16
get_course_files . . . . .	17
get_course_folders . . . . .	17
get_course_groups . . . . .	18
get_course_pages . . . . .	18
get_course_participation . . . . .	19
get_course_quizzes . . . . .	19
get_course_sections . . . . .	20
get_course_students . . . . .	20
get_department_grade_data . . . . .	21
get_department_participation_data . . . . .	21
get_department_statistics . . . . .	22
get_department_statistics_by_subaccount . . . . .	23
get_discussions . . . . .	23
get_favorite_courses . . . . .	24
get_group_categories . . . . .	24
get_group_info . . . . .	25
get_group_memberships . . . . .	25
get_group_users . . . . .	26
get_modules . . . . .	26
get_module_items . . . . .	27
get_page_content . . . . .	27
get_roles . . . . .	28
get_section_information . . . . .	28
get_student_summaries . . . . .	29
get_user_course_assignment_data . . . . .	29
get_user_course_messaging_data . . . . .	30
get_user_course_participation_data . . . . .	30
post_new_discussion . . . . .	31

---

canvas_api_key	<i>Get the Canvas API key from the environment variable</i>
----------------	---

---

**Description**

Get the Canvas API key from the environment variable

**Usage**

```
canvas_api_key()
```

**Value**

The Canvas API key stored in the CANVAS\_API\_KEY environment variable.

---

canvas_authenticate	<i>Authenticate with Canvas LMS API</i>
---------------------	---

---

**Description**

This function handles authentication with the Canvas LMS API. It uses the provided API key and base URL, or falls back to the CANVAS\_API\_KEY and CANVAS\_BASE\_URL environment variables if none are provided.

**Usage**

```
canvas_authenticate(api_key = canvas_api_key(), base_url = canvas_base_url())
```

**Arguments**

api_key	The API key for authenticating with the Canvas LMS API. Defaults to the CANVAS_API_KEY environment variable.
base_url	The base URL of the Canvas instance. Defaults to the CANVAS_BASE_URL environment variable.

**Value**

A list containing the authenticated 'api\_key' and 'base\_url'.

**Note**

The function verifies authentication by making a test request to the /api/v1/users/self endpoint of the Canvas instance. If the response status code is not 200, it throws an error message indicating that authentication failed.

**Examples**

```
## Not run:
# Authenticate with the Canvas LMS API
api_key <- "your_api_key"
base_url <- "https://canvas.example.com"
canvas <- canvas_authenticate(api_key, base_url)

## End(Not run)
```

---

canvas_base_url	<i>Get the Canvas base URL from the environment variable</i>
-----------------	--

---

**Description**

Get the Canvas base URL from the environment variable

**Usage**

```
canvas_base_url()
```

**Value**

The Canvas base URL stored in the CANVAS\_BASE\_URL environment variable.

---

create_assignment_group	<i>Create an Assignment Group in Canvas LMS</i>
-------------------------	---

---

**Description**

Creates a new assignment group in a specific course using the Canvas LMS API.

**Usage**

```
create_assignment_group(canvas, course_id, group_name, group_position = NULL)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which to create the assignment group.
group_name	The name of the assignment group.
group_position	The position of the assignment group in the course (optional).

**Value**

A confirmation message that the assignment group has been created.

---

`create_course_datalake`*Create a data lake for a course.*

---

**Description**

This function retrieves data from various endpoints for a specific course in the Canvas LMS API and stores the data as JSON files in a specified storage location.

**Usage**

```
create_course_datalake(canvas, course_id, storage_location)
```

**Arguments**

<code>canvas</code>	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
<code>course_id</code>	The ID of the course for which to create the data lake.
<code>storage_location</code>	The path to the storage location where the data files will be saved.

**Value**

NULL.

**Note**

This function retrieves data from various endpoints. Access to certain endpoints may require specific roles.

---

`create_course_section` *Create a Course Section in Canvas LMS*

---

**Description**

Creates a new course section in a specific course using the Canvas LMS API.

**Usage**

```
create_course_section(  
  canvas,  
  course_id,  
  section_name,  
  section_start_date = NULL,  
  section_end_date = NULL  
)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which to create the section.
section_name	The name of the section.
section_start_date	(Optional) The start date of the section.
section_end_date	(Optional) The end date of the section.

**Value**

A confirmation message that the section has been created.

---

create_folder	<i>Create a Folder in Canvas LMS</i>
---------------	--------------------------------------

---

**Description**

Creates a new folder in a specific course using the Canvas LMS API.

**Usage**

```
create_folder(canvas, course_id, folder_name, parent_folder_id = NULL)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which to create the folder.
folder_name	The name of the folder.
parent_folder_id	(Optional) The ID of the parent folder in which to create the folder.

**Value**

A confirmation message that the folder has been created.

---

create\_group\_category *Create a Group Category in Canvas LMS*

---

**Description**

Creates a new group category in a specific course using the Canvas LMS API.

**Usage**

```
create_group_category(  
    canvas,  
    course_id,  
    category_name,  
    allow_self_signup = FALSE  
)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which to create the group category.
category_name	The name of the group category.
allow_self_signup	(Optional) Whether to allow self-signup for groups in the category. Defaults to FALSE.

**Value**

A confirmation message that the group category has been created.

---

create\_page *Create a Page in Canvas LMS*

---

**Description**

Creates a new page in a specific course using the Canvas LMS API.

**Usage**

```
create_page(canvas, course_id, page_title, page_body, published = FALSE)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which to create the page.
page_title	The title of the page.
page_body	The body/content of the page.
published	(Optional) Whether the page should be published. Defaults to FALSE.

**Value**

A confirmation message that the page has been created.

---

delete\_course\_section *Delete a Course Section in Canvas LMS*

---

**Description**

Deletes an existing course section using the Canvas LMS API.

**Usage**

```
delete_course_section(canvas, section_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
section_id	The ID of the section to delete.

**Value**

A confirmation message that the section has been deleted.

---

download\_course\_file *Downloads a file from a given URL.*

---

**Description**

This function downloads a file from a specified URL and saves it locally.

**Usage**

```
download_course_file(canvas, file_url, download_path)
```



**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
file_url	The URL of the file to download.
download_path	The path where the file should be downloaded.

**Value**

The path of the downloaded file.

**Examples**

```
## Not run:  
# Download a file from a given URL  
canvas <- canvas_authenticate(api_key, base_url)  
file_url <- "https://example.com/file.pdf"  
download_path <- "path/to/save/file.pdf"  
file_path <- download_course_file(canvas, file_url, download_path)  
  
## End(Not run)
```

---

edit\_section

*Edit a Course Section in Canvas LMS*

---

**Description**

Modifies an existing course section using the Canvas LMS API.

**Usage**

```
edit_section(  
  canvas,  
  section_id,  
  section_name = NULL,  
  sis_section_id = NULL,  
  integration_id = NULL,  
  section_start_date = NULL,  
  section_end_date = NULL,  
  restrict_enrollments_to_section_dates = NULL,  
  override_sis_stickiness = NULL  
)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
section_id	The ID of the section to edit.
section_name	The new name of the section (optional).
sis_section_id	The new SIS ID of the section (optional).
integration_id	The new integration ID of the section (optional).
section_start_date	The new start date of the section (optional).
section_end_date	The new end date of the section (optional).
restrict_enrollments_to_section_dates	Whether to restrict user enrollments to the start and end dates of the section (optional).
override_sis_stickiness	Whether to override SIS stickiness (optional).

**Value**

A confirmation message that the section has been edited.

---

get_accounts	<i>Get a list of accounts from the Canvas LMS API</i>
--------------	---

---

**Description**

Retrieves a paginated list of accounts that the current user can view or manage.

**Usage**

```
get_accounts(canvas, include = NULL, per_page = 100)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
include	A vector of additional information to include. Default is NULL.
per_page	Number of accounts to retrieve per page. Default is 100.

**Value**

A list of accounts retrieved from the Canvas LMS API.

---

get_all_courses	<i>Retrieves a paginated list of all courses visible in the public index.</i>
-----------------	---

---

**Description**

This function retrieves a paginated list of all courses visible in the public index using the Canvas LMS API. *NOTE* This function might take a while to finish.

**Usage**

```
get_all_courses(canvas, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
per_page	(Optional) The number of courses to retrieve per page of results (default is 100).

**Value**

A data frame of courses visible in the public index.

---

get_assignments	<i>Get Assignments from Canvas LMS API</i>
-----------------	--

---

**Description**

Fetches a list of assignments within a specific course from the Canvas LMS API.

**Usage**

```
get_assignments(canvas, course_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the assignments.

**Value**

A list of assignments retrieved from the Canvas LMS API.

---

get\_assignment\_data     *Get course-level assignment data from the Canvas LMS API*

---

**Description**

Retrieves the course-level assignment data for a specific course from the Canvas LMS API.

**Usage**

```
get_assignment_data(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the assignment data.
per_page	Number of assignment data to retrieve per page. Default is 100.

**Value**

A list of assignment data retrieved from the Canvas LMS API.

---

get\_assignment\_details  
                          *Get Assignment Details from Canvas LMS API*

---

**Description**

Retrieves detailed information about a specific assignment from the Canvas LMS API.

**Usage**

```
get_assignment_details(canvas, course_id, assignment_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course in which the assignment resides.
assignment_id	The ID of the assignment for which to retrieve the details.

**Value**

A dataframe containing the detailed information about the assignment.

---

`get_assignment_submissions`*Lists assignment submissions for a course.*

---

**Description**

This function retrieves a list of assignment submissions for a specific course in the Canvas LMS API.

**Usage**

```
get_assignment_submissions(  
  canvas,  
  course_id,  
  assignment_id = NULL,  
  per_page = 100  
)
```

**Arguments**

<code>canvas</code>	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
<code>course_id</code>	The ID of the course for which to list assignment submissions.
<code>assignment_id</code>	(Optional) The ID of a specific assignment for which to list submissions.
<code>per_page</code>	(Optional) The number of submissions to retrieve per page of results (default is 100).

**Value**

A data frame of assignment submissions for the specified course and assignment.

---

`get_calendar_events`    *Get Calendar Events from Canvas LMS API*

---

**Description**

Retrieve the paginated list of calendar events or assignments for the current user.

**Usage**

```

get_calendar_events(
  canvas,
  type = "event",
  start_date = NULL,
  end_date = NULL,
  undated = FALSE,
  all_events = FALSE,
  context_codes = NULL,
  excludes = NULL,
  includes = NULL,
  important_dates = FALSE,
  blackout_date = FALSE
)

```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
type	The type of events to retrieve. Default is "event". Allowed values: event, assignment.
start_date	Only return events since the start_date (inclusive). Defaults to today.
end_date	Only return events before the end_date (inclusive). Defaults to start_date.
undated	Defaults to false (dated events only). If true, only return undated events and ignore start_date and end_date.
all_events	Defaults to false (uses start_date, end_date, and undated criteria). If true, all events are returned, ignoring start_date, end_date, and undated criteria.
context_codes	List of context codes of courses, groups, users, or accounts whose events you want to see. If not specified, defaults to the current user (i.e personal calendar, no course/group events).
excludes	Array of attributes to exclude. Possible values are "description", "child_events" and "assignment".
includes	Array of optional attributes to include. Possible values are "web_conference" and if calendar_series flag is on, "series_natural_language".
important_dates	Defaults to false. If true, only events with important dates set to true will be returned.
blackout_date	Defaults to false. If true, only events with blackout date set to true will be returned.

**Value**

A data frame of calendar events retrieved from the Canvas LMS API.

---

 get\_courses

*Get Courses from Canvas LMS API*


---

**Description**

Retrieves a list of courses from the Canvas LMS API.

**Usage**

```
get_courses(canvas, per_page = 100)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
per_page	Number of courses to retrieve per page. Default is 100.

**Value**

A list of courses retrieved from the Canvas LMS API.

---

 get\_course\_announcements

*Retrieves course announcements.*


---

**Description**

This function retrieves a list of announcements for a specific course in the Canvas LMS API.

**Usage**

```
get_course_announcements(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course for which to retrieve announcements.
per_page	(Optional) The number of announcements to retrieve per page of results (default is 100).

**Value**

A data frame of course announcements.

---

get\_course\_details      *Get Course Details from Canvas LMS API*

---

**Description**

Retrieves detailed information about a specific course from the Canvas LMS API.

**Usage**

```
get_course_details(canvas, course_id)
```

**Arguments**

canvas                    A list containing the 'api\_key' and 'base\_url' for authentication.  
course\_id                The ID of the course for which to retrieve the details.

**Value**

A dataframe containing the detailed information about the course.

---

get\_course\_enrollments      *Retrieves the course enrollments for a course.*

---

**Description**

This function retrieves the enrollments of students and other roles in a specific course in the Canvas LMS API.

**Usage**

```
get_course_enrollments(canvas, course_id, per_page = 100)
```

**Arguments**

canvas                    An object containing the Canvas API key and base URL, obtained through the canvas\_authenticate function.  
course\_id                The ID of the course for which to fetch the enrollments.  
per\_page                (Optional) The number of enrollments to retrieve per page of results (default is 100).

**Value**

A data frame of course enrollments for the specified course.



---

get_course_files	<i>Retrieves a list of files within a course.</i>
------------------	---

---

**Description**

This function retrieves a list of files within a specific course in the Canvas LMS API.

**Usage**

```
get_course_files(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
course_id	The ID of the course for which to fetch the files.
per_page	(Optional) The number of files to retrieve per page of results (default is 100).

**Value**

A data frame of files within the specified course.

---

get_course_folders	<i>Retrieves course folders.</i>
--------------------	----------------------------------

---

**Description**

This function retrieves a list of folders for a specific course in the Canvas LMS API.

**Usage**

```
get_course_folders(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
course_id	The ID of the course for which to retrieve folders.
per_page	Number of courses to retrieve per page. Default is 100.

**Value**

A data frame of course folders.

---

get\_course\_groups      *Retrieves the list of groups in a course.*

---

**Description**

This function retrieves the list of groups in a specific course in the Canvas LMS API.

**Usage**

```
get_course_groups(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to fetch the groups.
per_page	(Optional) The number of groups to retrieve per page of results (default is 100).

**Value**

A data frame of groups in the specified course.

---

get\_course\_pages      *Retrieves the pages within a course.*

---

**Description**

This function retrieves the pages within a specific course in the Canvas LMS API.

**Usage**

```
get_course_pages(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course for which to fetch the pages.
per_page	(Optional) The number of pages to retrieve per page of results (default is 50).

**Value**

A list of pages within the specified course.

---

`get_course_participation`*Get course-level participation data from Canvas LMS API*

---

**Description**

Fetches the participation data for a specific course from the Canvas LMS API.

**Usage**

```
get_course_participation(canvas, course_id)
```

**Arguments**

<code>canvas</code>	A list containing the 'api_key' and 'base_url' for authentication.
<code>course_id</code>	The ID of the course for which to retrieve the participation data.

**Value**

The participation data for the specified course retrieved from the Canvas LMS API.

---

`get_course_quizzes`     *Retrieves course quizzes.*

---

**Description**

This function retrieves a list of quizzes for a specific course in the Canvas LMS API.

**Usage**

```
get_course_quizzes(canvas, course_id, per_page = 100)
```

**Arguments**

<code>canvas</code>	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
<code>course_id</code>	The ID of the course for which to retrieve quizzes.
<code>per_page</code>	(Optional) The number of quizzes to retrieve per page of results (default is 100).

**Value**

A data frame of course quizzes.

---

get\_course\_sections    *Retrieves course sections.*

---

**Description**

This function retrieves a list of sections for a specific course in the Canvas LMS API.

**Usage**

```
get_course_sections(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course for which to retrieve sections.
per_page	(Optional) The number of sections to retrieve per page of results (default is 100).

**Value**

A data frame of course sections.

---

get\_course\_students    *Retrieves the list of students in a course.*

---

**Description**

This function retrieves the list of students enrolled in a specific course in the Canvas LMS API.

**Usage**

```
get_course_students(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course for which to fetch the students.
per_page	(Optional) The number of students to retrieve per page of results (default is 100).

**Value**

A data frame of students enrolled in the specified course.

---

`get_department_grade_data`*Get department-level grade data from the Canvas LMS API*

---

**Description**

Retrieves the department-level grade data for a specific account and term from the Canvas LMS API.

**Usage**

```
get_department_grade_data(  
    canvas,  
    account_id,  
    type = "current",  
    term_id = NULL,  
    per_page = 100  
)
```

**Arguments**

<code>canvas</code>	A list containing the 'api_key' and 'base_url' for authentication.
<code>account_id</code>	The ID of the account for which to retrieve the grade data.
<code>type</code>	The type of courses to include in the data. Can be 'current', 'completed', or 'term'.
<code>term_id</code>	The ID of the term for which to retrieve the grade data. Only used when type is 'terms/<term_id>'.
<code>per_page</code>	Number of grade data to retrieve per page. Default is 100.

**Value**

A data frame of grade data retrieved from the Canvas LMS API.

---

`get_department_participation_data`*Get department-level participation data from the Canvas LMS API*

---

**Description**

Retrieves the department-level participation data for a specific account and term from the Canvas LMS API.

**Usage**

```
get_department_participation_data(
    canvas,
    account_id,
    type = "current",
    term_id = NULL,
    per_page = 100
)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
account_id	The ID of the account for which to retrieve the participation data.
type	The type of courses to include in the data. Can be 'current', 'completed', or 'term'.
term_id	The ID of the term for which to retrieve the participation data. Only used when type is 'terms/<term_id>'.
per_page	Number of participation data to retrieve per page. Default is 100.

**Value**

A data frame of participation data retrieved from the Canvas LMS API.

---

```
get_department_statistics
```

*Get department-level statistics from the Canvas LMS API*

---

**Description**

Retrieves department-level statistics for a specific account and term from the Canvas LMS API.

**Usage**

```
get_department_statistics(canvas, account_id, type = "current", term_id = NULL)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
account_id	The ID of the account for which to retrieve the statistics.
type	The type of courses to include in the data. Can be 'current', 'completed', or 'term'.
term_id	The ID of the term for which to retrieve the statistics. Only used when type is 'terms/<term_id>'.

**Value**

A list of department-level statistics retrieved from the Canvas LMS API.

---

```
get_department_statistics_by_subaccount
    Get department-level statistics by subaccount from the Canvas LMS
    API
```

---

**Description**

Retrieves department-level statistics for a specific account and term from the Canvas LMS API.

**Usage**

```
get_department_statistics_by_subaccount(
    canvas,
    account_id,
    type = "current",
    term_id = NULL
)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
account_id	The ID of the account for which to retrieve the statistics.
type	The type of courses to include in the data. Can be 'current', 'completed', or 'term'.
term_id	The ID of the term for which to retrieve the statistics. Only used when type is 'terms/<term_id>'.

**Value**

A list of department-level statistics retrieved from the Canvas LMS API.

---

```
get_discussions    Retrieves the discussion topics within a course.
```

---

**Description**

This function retrieves the discussion topics within a specific course in the Canvas LMS API.

**Usage**

```
get_discussions(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the <code>canvas_authenticate</code> function.
course_id	The ID of the course for which to fetch the discussion topics.
per_page	The number of entries to show

**Value**

A list of discussion topics within the specified course.

---

`get_favorite_courses` *Get Favorite Courses in Canvas LMS*

---

**Description**

Retrieves the data of favorite courses for the authenticated user using the Canvas LMS API.

**Usage**

```
get_favorite_courses(canvas, exclude_blueprint_courses = NULL)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
exclude_blueprint_courses	When set, only return courses that are not configured as blueprint courses (optional).

**Value**

The dataframe of favorite courses.

---

`get_group_categories` *Get group categories for a context*

---

**Description**

This function retrieves the group categories for a specific context (e.g., course) in the Canvas LMS API.

**Usage**

```
get_group_categories(canvas, course_id)
```



**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the group categories.

**Value**

A data frame of group categories in the specified context.

---

get\_group\_info      *Get information about a single group*

---

**Description**

This function retrieves information about a specific group in the Canvas LMS API.

**Usage**

```
get_group_info(canvas, group_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
group_id	The ID of the group for which to retrieve the information.

**Value**

A list containing the information about the specified group.

---

get\_group\_memberships      *Get group memberships*

---

**Description**

This function retrieves the memberships for a specific group in the Canvas LMS API.

**Usage**

```
get_group_memberships(canvas, group_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
group_id	The ID of the group for which to retrieve the memberships.

**Value**

A data frame of memberships in the specified group.

---

get_group_users	<i>Get users in a group</i>
-----------------	-----------------------------

---

**Description**

This function retrieves the users in a specific group in the Canvas LMS API.

**Usage**

```
get_group_users(canvas, group_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
group_id	The ID of the group for which to retrieve the users.

**Value**

A data frame of users in the specified group.

---

get_modules	<i>Retrieves the modules within a course.</i>
-------------	---

---

**Description**

This function retrieves the modules within a specific course in the Canvas LMS API.

**Usage**

```
get_modules(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course for which to fetch the modules.
per_page	The number of entries to show

**Value**

A list of modules within the specified course.

---

get_module_items	<i>Retrieves the items within a specific module.</i>
------------------	--

---

**Description**

This function retrieves the items within a specific module of a course in the Canvas LMS API.

**Usage**

```
get_module_items(canvas, course_id, module_id)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course containing the module.
module_id	The ID of the module for which to fetch the items.

**Value**

A list of items within the specified module.

---

get_page_content	<i>Retrieves the content body of a specified page.</i>
------------------	--

---

**Description**

This function retrieves the content body of a specified page within a course in the Canvas LMS API.

**Usage**

```
get_page_content(canvas, course_id, page_id)
```

**Arguments**

canvas	An object containing the Canvas API key and base URL, obtained through the canvas_authenticate function.
course_id	The ID of the course to which the page belongs.
page_id	The ID of the page for which to fetch the content body.

**Value**

The content body of the specified page.

---

<code>get_roles</code>	<i>Retrieve Roles for a Canvas Account</i>
------------------------	--

---

**Description**

This function retrieves a paginated list of the roles available to a specific account in the Canvas LMS system.

**Usage**

```
get_roles(canvas, account_id, per_page = 100)
```

**Arguments**

<code>canvas</code>	A list containing the base URL and API key for the Canvas LMS instance.
<code>account_id</code>	The ID of the account for which you want to retrieve the roles.
<code>per_page</code>	The number of roles to retrieve per page (default is 100).

**Value**

A data frame containing the roles available to the specified account.

---

<code>get_section_information</code>	<i>Get Section Information in Canvas LMS</i>
--------------------------------------	--

---

**Description**

Retrieves information about a specific course section using the Canvas LMS API.

**Usage**

```
get_section_information(canvas, course_id, section_id)
```

**Arguments**

<code>canvas</code>	A list containing the 'api_key' and 'base_url' for authentication.
<code>course_id</code>	The ID of the course.
<code>section_id</code>	The ID of the section.

**Value**

The information about the section.

---

get\_student\_summaries *Get student summaries for a course from Canvas LMS API*

---

**Description**

Retrieves the student summaries for a specific course from the Canvas LMS API.

**Usage**

```
get_student_summaries(canvas, course_id, per_page = 100)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the student summaries.
per_page	Number of student summaries to retrieve per page. Default is 100.

**Value**

A list of student summaries retrieved from the Canvas LMS API.

---

get\_user\_course\_assignment\_data  
*Get user-in-a-course-level assignment data from the Canvas LMS API*

---

**Description**

Retrieves user-in-a-course-level assignment data for a specific course and student from the Canvas LMS API.

**Usage**

```
get_user_course_assignment_data(canvas, course_id, student_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the assignment data.
student_id	The ID of the student for which to retrieve the assignment data.

**Value**

A list of user-in-a-course-level assignment data retrieved from the Canvas LMS API.

---

get\_user\_course\_messaging\_data

*Get user-in-a-course-level messaging data from the Canvas LMS API*

---

**Description**

Retrieves user-in-a-course-level messaging data for a specific course and student from the Canvas LMS API.

**Usage**

```
get_user_course_messaging_data(canvas, course_id, student_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the messaging data.
student_id	The ID of the student for which to retrieve the messaging data.

**Value**

A list of user-in-a-course-level messaging data retrieved from the Canvas LMS API.

---

get\_user\_course\_participation\_data

*Get user-in-a-course-level participation data from the Canvas LMS API*

---

**Description**

Retrieves user-in-a-course-level participation data for a specific course and student from the Canvas LMS API.

**Usage**

```
get_user_course_participation_data(canvas, course_id, student_id)
```

**Arguments**

canvas	A list containing the 'api_key' and 'base_url' for authentication.
course_id	The ID of the course for which to retrieve the participation data.
student_id	The ID of the student for which to retrieve the participation data.

**Value**

A list of user-in-a-course-level participation data retrieved from the Canvas LMS API.

---

`post_new_discussion`    *Post a New Discussion in Canvas LMS*

---

**Description**

Creates a new discussion topic in a specific course using the Canvas LMS API.

**Usage**

```
post_new_discussion(  
  canvas,  
  course_id,  
  discussion_title,  
  discussion_message,  
  discussion_is_announcement = FALSE  
)
```

**Arguments**

<code>canvas</code>	A list containing the 'api_key' and 'base_url' for authentication.
<code>course_id</code>	The ID of the course in which to create the discussion.
<code>discussion_title</code>	The title of the discussion topic.
<code>discussion_message</code>	The initial message content of the discussion topic.
<code>discussion_is_announcement</code>	(Optional) Whether the discussion should be an announcement. Defaults to FALSE.

**Value**

A confirmation message that the discussion has been created.

# Index

canvas\_api\_key, 3  
canvas\_authenticate, 3  
canvas\_base\_url, 4  
create\_assignment\_group, 4  
create\_course\_datalake, 5  
create\_course\_section, 5  
create\_folder, 6  
create\_group\_category, 7  
create\_page, 7  
  
delete\_course\_section, 8  
download\_course\_file, 8  
  
edit\_section, 9  
  
get\_accounts, 10  
get\_all\_courses, 11  
get\_assignment\_data, 12  
get\_assignment\_details, 12  
get\_assignment\_submissions, 13  
get\_assignments, 11  
get\_calendar\_events, 13  
get\_course\_announcements, 15  
get\_course\_details, 16  
get\_course\_enrollments, 16  
get\_course\_files, 17  
get\_course\_folders, 17  
get\_course\_groups, 18  
get\_course\_pages, 18  
get\_course\_participation, 19  
get\_course\_quizzes, 19  
get\_course\_sections, 20  
get\_course\_students, 20  
get\_courses, 15  
get\_department\_grade\_data, 21  
get\_department\_participation\_data, 21  
get\_department\_statistics, 22  
get\_department\_statistics\_by\_subaccount,  
23  
get\_discussions, 23  
  
get\_favorite\_courses, 24  
get\_group\_categories, 24  
get\_group\_info, 25  
get\_group\_memberships, 25  
get\_group\_users, 26  
get\_module\_items, 27  
get\_modules, 26  
get\_page\_content, 27  
get\_roles, 28  
get\_section\_information, 28  
get\_student\_summaries, 29  
get\_user\_course\_assignment\_data, 29  
get\_user\_course\_messaging\_data, 30  
get\_user\_course\_participation\_data, 30  
  
post\_new\_discussion, 31