Package ‘vvcanvas’

July 17, 2023

Title 'Canvas' LMS API Integration

Version 0.0.2

Description
Allow R users to interact with the 'Canvas' Learning Management System (LMS) API (see <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 2023-07-17 07:20:02 UTC

R topics documented:
canvas_authenticate .................................................. 2
create_assignment_group ........................................... 3
create_course_datalake ............................................. 3
create_course_section .............................................. 4
create_folder .......................................................... 5
create_group_category ............................................. 5
canvas_authenticate

Authenticate with Canvas LMS API

Description

Handles authentication with the Canvas LMS API using the provided API key and base URL.

Usage

canvas_authenticate(api_key, base_url)

Arguments

api_key The API key for authenticating with the Canvas LMS API.
base_url The base URL of the Canvas instance.

Value

A list containing the authenticated 'api_key' and 'base_url'.
create_assignment_group

## Not run:

```r
# 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)

---

### Description

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

### Usage

```r
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

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

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 create the data lake.
- **storage_location**: 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**

```python
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  
Create a Folder in Canvas LMS

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.


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 canvas_authenticate function.
file_url: The URL of the file to download.
download_path: The path where the file should be downloaded.
get_all_courses

Value

The path of the downloaded file.

Examples

```r
## 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)
```

get_all_courses

Retrieves a paginated list of all courses visible in the public index.

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

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

Arguments

- **canvas**: An object containing the Canvas API key and base URL, obtained through the `canvas_authenticate` 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  
*Get Assignments from Canvas LMS API*

**Description**

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

**Usage**

```python
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_details  
*Get Assignment Details from Canvas LMS API*

**Description**

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

**Usage**

```python
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

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

Value

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>canvas</td>
<td>A list containing the 'api_key' and 'base_url' for authentication.</td>
</tr>
<tr>
<td>per_page</td>
<td>Number of courses to retrieve per page. Default is 100.</td>
</tr>
</tbody>
</table>
get_course_details

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

Retrieves a list of files within a course.

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 canvas_authenticate 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*  
Retrieves course folders.

**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 `canvas_authenticate` 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**
```r
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_quizzes**  
Retrieves course quizzes.

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

**Usage**
```r
get_course_quizzes(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 quizzes.
- **per_page**: (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_discussions**

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 `canvas_authenticate` 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_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**

*Get users in a group*

**Description**

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

**Usage**

```r
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**

*Retrieves the modules within a course.*

**Description**

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

**Usage**

```r
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  
Retrieves the items within a specific module.

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  
Retrieves the content body of a specified page.

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.
### 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**

```r
def post_new_discussion(
    canvas,
    course_id,
    discussion_title,
    discussion_message,
    discussion_is_announcement = 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 discussion.
- `discussion_title`: The title of the discussion topic.
- `discussion_message`: The initial message content of the discussion topic.
- `discussion_is_announcement`: (Optional) Whether the discussion should be an announcement. Defaults to FALSE.

**Value**

A confirmation message that the discussion has been created.
Index

canvas_authenticate, 2
create_assignment_group, 3
create_course_datalake, 3
create_course_section, 4
create_folder, 5
create_group_category, 5
create_page, 6
download_course_file, 6
get_all_courses, 7
get_assignment_details, 8
get_assignment_submissions, 9
get_assignments, 8
get_course_announcements, 10
get_course_details, 10
get_course_enrollments, 11
get_course_files, 11
get_course_folders, 12
get_course_groups, 12
get_course_pages, 13
get_course_quizzes, 13
get_course_sections, 14
get_course_students, 14
get_courses, 9
get_discussions, 15
get_group_categories, 15
get_group_info, 16
get_group_memberships, 16
get_group_users, 17
get_module_items, 18
get_modules, 17
get_page_content, 18
post_new_discussion, 19