canvasapi Documentation

Release 0.13.0

University of Central Florida (Center for Distributed Learning)

Jul 17, 2019

Contents

Contents:

CHAPTER 1

Getting Started with CanvasAPI

1.1 Installing CanvasAPI

You can install CanvasAPI with pip:

```
pip install canvasapi
```

1.2 Usage

Before using CanvasAPI, you'll need to instantiate a new Canvas object:

```
# Import the Canvas class
from canvasapi import Canvas

# Canvas API URL
API_URL = "https://example.com"
# Canvas API key
API_KEY = "p@$$w0rd"

# Initialize a new Canvas object
canvas = Canvas(API_URL, API_KEY)
```

You can now use `canvas` to make API calls.

1.3 Working with Canvas Objects

CanvasAPI converts the JSON responses from the Canvas API into Python objects. These objects provide further access to the Canvas API. You can find a full breakdown of the methods these classes provide in our class documentation. Below, you'll find a few examples of common CanvasAPI use cases.

1.3.1 Course objects

Courses can be retrieved from the API:

```
# Grab course 123456
>>> course = canvas.get_course(123456)

# Access the course's name
>>> course.name
'Test Course'

# Update the course's name
>>> course.update(course={'name': 'New Course Name'})
```

See our documentation on *keyword arguments* for more information about how `course.update()` handles the `name` argument.

1.3.2 User objects

```
# Grab user 123
>>> user = canvas.get_user(123)

# Access the user's name
>>> user.name
'Test User'

# Retrieve a list of courses the user is enrolled in
>>> courses = user.get_courses()

# Grab a different user by their SIS ID
>>> login_id_user = canvas.get_user('some_user', 'sis_login_id')
```

1.3.3 Paginated Lists

Some calls, like the `user.get_courses()` call above, will request multiple objects from Canvas's API. CanvasAPI collects these objects in a `PaginatedList` object. `PaginatedList` generally acts like a regular Python list. You can grab an element by index, iterate over it, and take a slice of it.

Warning: `PaginatedList` lazily loads its elements. Unfortunately, there's no way to determine the exact number of records Canvas will return without traversing the list fully. This means that `PaginatedList` isn't aware of its own length and negative indexing is not currently supported.

Let's look at how we can use the `PaginatedList` returned by our `get_courses()` call:

```
# Retrieve a list of courses the user is enrolled in
>>> courses = user.get_courses()

>>> print(courses)
<PaginatedList of type Course>

# Access the first element in our list.
#
# You'll notice the first call takes a moment, but the next N-1
# elements (where N = the per_page argument supplied; the default is 10)
# will be instantly accessible.
```

(continues on next page)

(continued from previous page)

```
>>> print(courses[0])
TST101 Test Course (1234567)

# Iterate over our course list
>>> for course in courses:
         print(course)

TST101 Test Course 1 (1234567)
TST102 Test Course 2 (1234568)
TST103 Test Course 3 (1234569)

# Take a slice of our course list
>>> courses[:2]
[TST101 Test Course 1 (1234567), TST102 Test Course 2 (1234568)]
```

1.3.4 Keyword Arguments

Most of Canvas's API endpoints accept a variety of arguments. CanvasAPI allows developers to insert keyword arguments when making calls to endpoints that accept arguments.

```
# Get all of the active courses a user is currently enrolled in
>>> courses = user.get_courses(enrollment_status='active')

# Fetch 50 objects per page when making calls that return a PaginatedList
>>> courses = user.get_courses(per_page=50)
```

For a more detailed description of how CanvasAPI handles more complex keyword arguments, check out the Keyword Argument Documentation.

CHAPTER 2

Examples

Here are some examples of common tasks that can be performed easily and quickly with CanvasAPI.

2.1 Boilerplate

All other examples assume you have already created a `canvas` object like so:

```python
# Import the Canvas class
from canvasapi import Canvas

# Canvas API URL
API_URL = "https://example.com"
# Canvas API key
API_KEY = "p@$$w0rd"

# Initialize a new Canvas object
canvas = Canvas(API_URL, API_KEY)
```

2.2 Accounts

2.2.1 Creating a New User

```python
# Grab the account to create the user under
account = canvas.get_account(1)

user = account.create_user(
    user={
        'name': 'New User'
    },
    pseudonym={
```

(continues on next page)

(continued from previous page)

```
        'password': 'secure123',
        'sis_user_id': 'new_user'
    }
)
```

2.2.2 List Courses under an Account

```
courses = account.get_courses()

for course in courses:
    print(course)
```

2.3 Users

2.3.1 Get a User by SIS ID

```
# Grab a user with the SIS ID of 'some_id'
user = canvas.get_user('some_id', 'sis_login_id')
```

2.3.2 Get a user by their Canvas ID

```
# Grab user with ID of 1
user = canvas.get_user(1)
```

2.3.3 Edit an Existing User

```
# Grab the user
user = canvas.get_user(1)

user.edit(
    user={'name': 'New Name'}
)
```

2.3.4 Get a User's Page Views

```
page_views = user.get_page_views()
```

2.4 Logins

2.4.1 List User Logins

```
logins = user.get_user_logins()

for login in logins:
    print(login)
```

2.5 Courses

2.5.1 Get a Course by Canvas ID

```
# Grab course with ID of 123456
course = canvas.get_course(123456)
```

2.5.2 Get Users in a Course

Using *canvasapi.course.Course.get_users()*:

Get All Users

```
users = course.get_users()

for user in users:
    print(user)
```

Get Only Students

```
users = course.get_users(enrollment_type=['student'])

for user in users:
    print(user)
```

Get Only Teachers, TAs, and Designers

```
type_list = ['teacher', 'ta', 'designer']

users = course.get_users(enrollment_type=type_list)

for user in users:
    print(user)
```

Get Only Active and Invited Students And Teachers

```
users = course.get_users(
    enrollment_type=['teacher', 'student']
    enrollment_state=['active', 'invited']
)

for user in users:
    print(user)
```

2.5.3 Update (Edit) a Course

Using `canvasapi.course.Course.update()`:

Update Course Name

```
print(course.name)  # prints 'Old Name'

course.update(course={'name': 'New Name'})

print(course.name)  # prints 'New Name'
```

Update Course Start and End Dates

Either an ISO8601 format date string:

```
course.update(
    course={
        'start_at': '2018-01-01T00:01Z',
        'end_at': '2018-12-31T11:59Z'
    }
)
```

Or a Python DateTime object:

```
from datetime import datetime

start_date = datetime(2018, 1, 1, 0, 1)
end_date = datetime(2018, 12, 31, 11, 59)

course.update(
    course={
        'start_at': start_date,
        'end_at': end_date
    }
)
```

2.5.4 Conclude a Course

Using `canvasapi.course.Course.conclude()`:

```
course.conclude()
```

2.5.5 Delete a Course

Using *canvasapi.course.Course.delete()*:

```
course.delete()
```

2.6 Assignments

2.6.1 Get a Single Assignment

Using *canvasapi.course.Course.get_assignment()*:

```
# Grab assignment with ID of 1234
assignment = course.get_assignment(1234)

print(assignment)
```

2.6.2 Get Multiple Assignments

Using *canvasapi.course.Course.get_assignments()*:

Get All Assignments

```
assignments = course.get_assignments()

for assignment in assignments:
    print(assignment)
```

Get Ungraded Assignments

```
assignments = course.get_assignments(bucket='ungraded')

for assignment in assignments:
    print(assignment)
```

2.6.3 Create an Assignment

Using *canvasapi.course.Course.create_assignment()*:

Create a Basic Assignment

```
new_assignment = course.create_assignment({
    'name': 'Assignment 1'
})

print(new_assignment)
```

Create an Assignment with Multiple Submission Types

```
new_assignment = course.create_assignment({
    'name': 'Assignment 2',
    'submission_types': ['online_upload', 'online_text_entry', 'online_url']
})

print(new_assignment)
```

Create a Complex Assignment

```
from datetime import datetime

new_assignment = course.create_assignment({
    'name': 'Assignment 3',
    'submission_types': ['online_upload'],
    'allowed_extensions': ['docx', 'doc', 'pdf'],
    'notify_of_update': True,
    'points_possible': 100,
    'due_at': datetime(2018, 12, 31, 11, 59),
    'description': 'Please upload your homework as a Word document or PDF.',
    'published': True
})

print(new_assignment)
```

2.6.4 Update (Edit) an Assignment

Using *canvasapi.assignment.Assignment.edit()*:

Update an Assignment's Name

```
updated_assignment = assignment.edit(assignment={'name': 'New Name'})

print(updated_assignment)
```

Change an Assignment's Submission Type(s)

```
updated_assignment = assignment.edit(
    assignment={'submission_types': ['on_paper']}
)

print(updated_assignment)
```

2.6.5 Delete an Assignment

Using *canvasapi.assignment.Assignment.delete()*:

```
assignment.delete()
```

CHAPTER 3

Keyword Arguments

3.1 Basic Parameters

In general, basic parameters can be passed directly as keyword arguments.

For example, *canvasapi.course.Course.get_users()* has several basic parameters including `search_term` and `user_id`, as shown in the Canvas Docs for List Users in Course.

Given an existing *canvasapi.course.Course* object, `course`, the parameter `search_term` can be passed like this:

```
course.get_users(search_term='John Doe')
```

Multiple basic arguments can be passed. In this case, `enrollment_role` has also been added.

```
course.get_users(
    search_term='John Doe',
    enrollment_role='StudentEnrollment'
)
```

3.2 List Parameters

Some endpoints have parameters that are designed to be passed a list. These usually look like `foo[]`.

For example, *canvasapi.course.Course.get_users()* has a few list parameters: `enrollment_type[]`, `include[]`, `user_ids[]`, and `enrollment_state[]`, as shown in the Canvas Docs for List Users in Course. To use these parameters, just pass a list to the keyword. CanvasAPI will automatically detect the list and convert the parameter to the right format. For instance, `enrollment_type[]` can be passed like this:

```
course.get_users(enrollment_type=['teacher', 'student'])
```

Multiple list parameters can be passed, including in combination with basic parameters. For this example, `include[]` and `search_term` have been added. Note that even though only one option was sent to `include[]`, it is still a list.

```
course.get_users(
    enrollment_type=['teacher', 'student'],
    search_term='John',
    include=['email']
)
```

3.3 Nested Parameters

Some endpoints have parameters that look like `foo[bar]`. Typically, there will be multiple parameters with the same prefix in the same endpoint.

For example, *`canvasapi.account.Account.create_course()`* has several parameters that look like `course[foo]`, as shown in the Canvas Docs for Create a New Course. However, square brackets are not valid characters for Python variables, so the following would **NOT** work:

```
# This is not valid, and will not work.
account.create_course(course[name]='Example Course')
```

What Canvas is effectively doing with the bracket format is grouping things into objects. To achieve a similar effect in Python, CanvasAPI uses dictionaries.

Given an existing *`canvasapi.account.Account`* object, `account`, the parameter `course[name]` can be passed like this:

```
account.create_course(course={'name': 'Example Course'})
```

In the background, CanvasAPI will combine the keys of the dictionary with the keyword of the argument, and ultimately send the correct variable to Canvas.

The benefit of this pattern is that multiple parameters with the same prefix can be sent to the same keyword argument. So to pass the `course[name]`, `course[course_code]`, and `course[is_public]` arguments, it would look like this:

```
account.create_course(
    course={
        'name': 'Example Course',
        'course_code': 'TST1234',
        'is_public': True
    }
)
```

Nested parameters work easily alongside basic (and list) parameters. For example, `offer` and `enroll_me`:

```
account.create_course(
    course={
        'name': 'Example Course',
        'course_code': 'TST1234',
        'is_public': True
    },
    enroll_me=True,
    offer=False
)
```

3.4 Complex Parameters

The three main types of parameters (basic, list, and nested) from above cover most types of parameters Canvas expects. However, there are some types that look deceptively more complex than they actually are. Below are some examples of how to handle these in CanvasAPI.

3.4.1 Deep Nested Parameters

canvasapi.user.User.edit() has the parameter `user[avatar][url]`, as shown in the Canvas Docs for Edit a User. Any parameter that takes the form of `foo[bar1][bar2]` with multiple bracketed sub-parameters follows the same rules as normal nested parameters, but with deeper nesting.

```
user.edit(
    user={
        'avatar': {
            'url': 'http://example.com/john_avatar.png'
        }
    }
)
```

3.4.2 List of Nested Parameters

canvasapi.account.Account.add_grading_standards() has the parameters `grading_scheme_entry[][name]` and `grading_scheme_entry[][value]`, as shown in the Canvas Docs for Create a New Grading Standard. Any parameter that takes the form of `foo[][bar]` can be represented by a list of dictionaries.

```
account.add_grading_standards(
    title='New Grading Standard',
    grading_scheme_entry=[
        {
            'name': 'A',
            'value': 90
        },
        {
            'name': 'B',
            'value': 80
        }
    ]
)
```

3.4.3 Nested List Parameters

canvasapi.course.Course.create_assignment() has the parameters `assignment[submission_types][]` and `assignment[allowed_extensions][]`, as shown in the Canvas Docs for Create an Assignment. Any parameter that takes the form of `foo[bar][]` is a nested parameter of which the value is a list.

```
course.create_assignment(
    assignment={
        'name': 'Assignment 1',
```

(continues on next page)

(continued from previous page)

```
        'submission_types': ['online_text_entry', 'online_upload'],
        'allowed_extensions': ['doc', 'docx']
    }
)
```

CHAPTER 4

Exceptions

CanvasAPI may return a number of different exceptions, which are listed below.

4.1 Quick Guide

Exception	Status Code	Explanation
`BadRequest`	400	Canvas was unable to process the request.
`InvalidAccessToken`	401	The supplied API key is invalid.
`Unauthorized`	401	CanvasAPI's key is valid, but is unauthorized to access the requested resource.
`Forbidden`	403	Canvas has denied access to the resource for this user.
`ResourceDoesNotExist`	404	Canvas could not locate the requested resource.
`Conflict`	409	Canvas had a conflict with an existing resource.
`RequiredFieldMissing`	N/A	A required keyword argument was not included.
`CanvasException`	N/A	An unknown error was thrown.

4.2 Class Reference

class canvasapi.exceptions.**CanvasException**(*message*)

Base class for all errors returned by the Canvas API.

The `CanvasException` exception is a basic library exception that all other exceptions inherit from. It is also thrown whenever an error occurs but a more specific exception isn't available or appropriate.

Here's a simple example of catching a `CanvasException`:

```
from canvasapi.exceptions import CanvasException
```

(continues on next page)

(continued from previous page)

```
try:
    canvas.get_course(1)
except CanvasException as e:
    print(e)
```

class canvasapi.exceptions.**BadRequest**(*message*)

Canvas was unable to understand the request. More information may be needed.

The *BadRequest* exception is thrown when Canvas returns an HTTP 400 error.

class canvasapi.exceptions.**InvalidAccessToken**(*message*)

CanvasAPI was unable to make an API connection.

The *InvalidAccessToken* exception is thrown when Canvas returns an HTTP 401 error and includes a `WWW-Authenticate` header.

This indicates that the supplied API Key is invalid.

class canvasapi.exceptions.**Unauthorized**(*message*)

CanvasAPI's key is valid, but is unauthorized to access the requested resource.

The *Unauthorized* exception is thrown when Canvas returns an HTTP 401 error and does **NOT** include a `WWW-Authenticate` header.

This indicates that the supplied API Key is probably valid, but the calling user does not have permission to access this resource.

class canvasapi.exceptions.**ResourceDoesNotExist**(*message*)

Canvas could not locate the requested resource.

The *ResourceDoesNotExist* exception is thrown when Canvas returns an HTTP 404 error.

class canvasapi.exceptions.**RequiredFieldMissing**(*message*)

A required field is missing.

The *RequiredFieldMissing* exception is thrown when required fields are not passed to a method's keyword arguments. This is common in cases where the required field must be represented as a dictionary. See our documentation on keyword arguments for examples of how to use keyword arguments in CanvasAPI.

class canvasapi.exceptions.**Forbidden**(*message*)

Canvas has denied access to the resource for this user.

The *Forbidden* exception is thrown when Canvas returns an HTTP 403 error.

class canvasapi.exceptions.**Conflict**(*message*)

Canvas had a conflict with an existing resource.

The *Conflict* exception is thrown when Canvas returns an HTTP 409 error.

CHAPTER 5

Debugging

As you work with CanvasAPI, you may encounter responses that don't match your expectations. In some cases, fixing the discrepancy is as simple as realizing you've misspelled a parameter name. In others, the CanvasAPI library might be improperly handling your input, and you'll need to submit a pull request or a bug report to get it resolved. Sometimes Canvas itself may be responding erroneously.

5.1 Capturing Logs

CanvasAPI emits logs for important events through the standard logging module. Inspecting these logs can make debugging much easier, as they provide a closer look at the abstracted network activity of the library. In order to view them, you'll need to tell Python where to put them.

A basic logging configuration might look like this:

```
import logging
import sys

logger = logging.getLogger("canvasapi")
handler = logging.StreamHandler(sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')

handler.setLevel(logging.DEBUG)
handler.setFormatter(formatter)
logger.addHandler(handler)
logger.setLevel(logging.DEBUG)
```

Once configured, any log event in the `canvasapi` module namespace will be printed to `sys.stdout`:

```
# A previously configured `Canvas` client
>>> canvas.get_current_user()
2019-07-08 14:22:24,517 - canvasapi.requester - INFO - Request: GET https://base/api/
↪v1/users/self
2019-07-08 14:22:24,517 - canvasapi.requester - DEBUG - Headers: {'Authorization':
↪'****4BSt'}
```

(continues on next page)

(continued from previous page)

```
2019-07-08 14:22:24,748 - canvasapi.requester - INFO - Response: GET https://base/api/
↪v1/users/self 200
2019-07-08 14:22:24,749 - canvasapi.requester - DEBUG - Headers: {'Cache-Control':
↪'max-age=0, private, must-revalidate',
'Connection': 'keep-alive',
'Content-Encoding': 'gzip',
'Content-Length': '329',
'Content-Type': 'application/json; charset=utf-8',
2019-07-08 14:22:24,749 - canvasapi.requester - DEBUG - Data: {'avatar_url': 'https://
↪base/images/thumbnails/43244/Umo5dyAg0OS3tpDtDN',
'created_at': '2014-08-22T08:02:00-04:00',
'effective_locale': 'en',
'email': '',
'id': XXXX181,
'integration_id': None,
'locale': None,
'login_id': 'XXXXXXXX181',
'name': 'Some User',
'permissions': {'can_update_avatar': True, 'can_update_name': False},
'root_account': 'xxxxxx.edu',
'short_name': 'Some User',
'sis_user_id': 'XXXXXXXX181',
'sortable_name': 'User S'}
```

CHAPTER 6

Class Reference

6.1 Canvas

class canvasapi.**Canvas**(*base_url*, *access_token*)

The main class to be instantiated to provide access to Canvas's API.

Parameters

- **base_url** (*str*) – The base URL of the Canvas instance's API.
- **access_token** (*str*) – The API key to authenticate requests with.

clear_course_nicknames()

Remove all stored course nicknames.

Calls DELETE /api/v1/users/self/course_nicknames

Returns True if the nicknames were cleared, False otherwise.

Return type bool

conversations_batch_update(*conversation_ids*, *event*)

Calls PUT /api/v1/conversations

Parameters

- **conversation_ids** (*list* of *str*) – List of conversations to update. Limited to 500 conversations.
- **event** (*str*) – The action to take on each conversation.

Return type *canvasapi.progress.Progress*

conversations_get_running_batches()

Returns any currently running conversation batches for the current user. Conversation batches are created when a bulk private message is sent asynchronously.

Calls GET /api/v1/conversations/batches

Returns dict with list of batch objects - not currently a Class

Return type *dict*

conversations_mark_all_as_read()

Mark all conversations as read.

Calls POST /api/v1/conversations/mark_all_as_read

Return type *bool*

conversations_unread_count()

Get the number of unread conversations for the current user

Calls GET /api/v1/conversations/unread_count

Returns simple object with unread_count, example: {'unread_count': '7'}

Return type *dict*

create_account(***kwargs*)

Create a new root account.

Calls POST /api/v1/accounts

Return type *canvasapi.account.Account*

create_appointment_group(*appointment_group*, ***kwargs*)

Create a new Appointment Group.

Calls POST /api/v1/appointment_groups

Parameters

- **appointment_group** (*dict*) – The attributes of the appointment group.
- **title** (*str*) – The title of the appointment group.

Return type *canvasapi.appointment_group.AppointmentGroup*

create_calendar_event(*calendar_event*, ***kwargs*)

Create a new Calendar Event.

Calls POST /api/v1/calendar_events

Parameters **calendar_event** (*dict*) – The attributes of the calendar event.

Return type *canvasapi.calendar_event.CalendarEvent*

create_conversation(*recipients*, *body*, ***kwargs*)

Create a new Conversation.

Calls POST /api/v1/conversations

Parameters

- **recipients** (*list of str*) – An array of recipient ids. These may be user ids or course/group ids prefixed with 'course_' or 'group_' respectively, e.g. recipients=['1', '2', 'course_3']
- **body** (*str*) – The body of the message being added.

Return type list of *canvasapi.conversation.Conversation*

create_group(***kwargs*)

Create a group

Calls POST /api/v1/groups/

Return type *canvasapi.group.Group*

create_planner_note(***kwargs*)
Create a planner note for the current user

Calls POST /api/v1/planner_notes

Return type *canvasapi.planner.PlannerNote*

create_planner_override(*plannable_type*, *plannable_id*, ***kwargs*)
Create a planner override for the current user

Calls POST /api/v1/planner/overrides

Parameters

- **plannable_type** (*str*) – Type of the item that you are overriding in the planner
- **plannable_id** (int or *canvasapi.planner.PlannerOverride*) – ID of the item that you are overriding in the planner

Return type *canvasapi.planner.PlannerOverride*

create_poll(*poll*, ***kwargs*)
Create a new poll for the current user.

Calls POST /api/v1/polls

Parameters **poll** (*list of dict*) – List of polls to create. *'question'* key is required.

Return type *canvasapi.poll.Poll*

get_account(*account*, *use_sis_id=False*, ***kwargs*)
Retrieve information on an individual account.

Calls GET /api/v1/accounts/:id

Parameters

- **account** (int, str or *canvasapi.account.Account*) – The object or ID of the account to retrieve.
- **use_sis_id** (*bool*) – Whether or not account_id is an sis ID. Defaults to *False*.

Return type *canvasapi.account.Account*

get_accounts(***kwargs*)
List accounts that the current user can view or manage.

Typically, students and teachers will get an empty list in response. Only account admins can view the accounts that they are in.

Calls GET /api/v1/accounts

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.account.Account*

get_activity_stream_summary()
Return a summary of the current user's global activity stream.

Calls GET /api/v1/users/self/activity_stream/summary

Return type dict

get_announcements(***kwargs*)
List announcements.

Calls GET /api/v1/announcements

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.discussion_topic.DiscussionTopic`*

`get_appointment_group`(*appointment_group*)
Return single Appointment Group by id

Calls GET /api/v1/appointment_groups/:id

Parameters **`appointment_group`** (*`canvasapi.appointment_group.AppointmentGroup`* or int) – The ID of the appointment group.

Return type *`canvasapi.appointment_group.AppointmentGroup`*

`get_appointment_groups`(***kwargs*)
List appointment groups.

Calls GET /api/v1/appointment_groups

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.appointment_group.AppointmentGroup`*

`get_calendar_event`(*calendar_event*)
Return single Calendar Event by id

Calls GET /api/v1/calendar_events/:id

Parameters **`calendar_event`** (*`canvasapi.calendar_event.CalendarEvent`* or int) – The object or ID of the calendar event.

Return type *`canvasapi.calendar_event.CalendarEvent`*

`get_calendar_events`(***kwargs*)
List calendar events.

Calls GET /api/v1/calendar_events

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.calendar_event.CalendarEvent`*

`get_conversation`(*conversation*, ***kwargs*)
Return single Conversation

Calls GET /api/v1/conversations/:id

Parameters **`conversation`** (*`canvasapi.conversation.Conversation`* or int) – The object or ID of the conversation.

Return type *`canvasapi.conversation.Conversation`*

`get_conversations`(***kwargs*)
Return list of conversations for the current user, most resent ones first.

Calls GET /api/v1/conversations

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.conversation.Conversation`*

`get_course`(*course*, *use_sis_id=False*, ***kwargs*)
Retrieve a course by its ID.

Calls GET /api/v1/courses/:id

Parameters

- **`course`** (int, str or *`canvasapi.course.Course`*) – The object or ID of the course to retrieve.

- **use_sis_id** (*bool*) – Whether or not course_id is an sis ID. Defaults to *False*.

Return type *canvasapi.course.Course*

get_course_accounts()

List accounts that the current user can view through their admin course enrollments (Teacher, TA or designer enrollments).

Only returns *id*, *name*, *workflow_state*, *root_account_id* and *parent_account_id*.

Calls GET /api/v1/course_accounts

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.account.Account*

get_course_nickname(*course*)

Return the nickname for the given course.

Calls GET /api/v1/users/self/course_nicknames/:course_id

Parameters **course** (*canvasapi.course.Course* or int) – The object or ID of the course.

Return type *canvasapi.course.CourseNickname*

get_course_nicknames()

Return all course nicknames set by the current account.

Calls GET /api/v1/users/self/course_nicknames

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.course.CourseNickname*

get_courses(***kwargs*)

Return a list of active courses for the current user.

Calls GET /api/v1/courses

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.course.Course*

get_epub_exports(***kwargs*)

Return a list of epub exports for the associated course.

Calls GET /api/v1/epub_exports

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.course_epub_export.CourseEpubExport*

get_file(*file*, ***kwargs*)

Return the standard attachment json object for a file.

Calls GET /api/v1/files/:id

Parameters **file** (*canvasapi.file.File* or int) – The object or ID of the file to retrieve.

Return type *canvasapi.file.File*

get_folder(*folder*)

Return the details for a folder

Calls GET /api/v1/folders/:id

Parameters **folder** (*canvasapi.folder.Folder* or int) – The object or ID of the folder to retrieve.

Return type *canvasapi.folder.Folder*

get_group(*group*, *use_sis_id=False*, ***kwargs*)

Return the data for a single group. If the caller does not have permission to view the group a 401 will be returned.

Calls GET /api/v1/groups/:group_id

Parameters

- **group** (*canvasapi.group.Group* or int) – The object or ID of the group to get.
- **use_sis_id** (*bool*) – Whether or not group_id is an sis ID. Defaults to *False*.

Return type *canvasapi.group.Group*

get_group_category(*category*)

Get a single group category.

Calls GET /api/v1/group_categories/:group_category_id

Parameters category (*canvasapi.group.GroupCategory* or int) – The object or ID of the category.

Return type *canvasapi.group.GroupCategory*

get_group_participants(*appointment_group*, ***kwargs*)

List student group participants in this appointment group.

Calls GET /api/v1/appointment_groups/:id/groups

Parameters appointment_group (*canvasapi.appointment_group.AppointmentGroup* or int) – The object or ID of the appointment group.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.group.Group*

get_outcome(*outcome*)

Returns the details of the outcome with the given id.

Calls GET /api/v1/outcomes/:id

Parameters outcome (*canvasapi.outcome.Outcome* or int) – The outcome object or ID to return.

Returns An Outcome object.

Return type *canvasapi.outcome.Outcome*

get_outcome_group(*group*)

Returns the details of the Outcome Group with the given id.

Calls GET /api/v1/global/outcome_groups/:id

Parameters group (*canvasapi.outcome.OutcomeGroup* or int) – The outcome group object or ID to return.

Returns An outcome group object.

Return type *canvasapi.outcome.OutcomeGroup*

get_planner_note(*planner_note*, ***kwargs*)

Retrieve a planner note for the current user

Calls GET /api/v1/planner_notes/:id

Parameters planner_note (int or *canvasapi.planner.PlannerNote*) – The ID of the planner note to retrieve.

Return type *canvasapi.planner.PlannerNote*

get_planner_notes(***kwargs*)
Retrieve the paginated list of planner notes

Calls GET /api/v1/planner_notes

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.planner.PlannerNote*

get_planner_override(*planner_override*, ***kwargs*)
Retrieve a planner override for the current user

Calls GET /api/v1/planner/overrides/:id

Parameters **planner_override** (int or *canvasapi.planner.PlannerOverride*) – The override or the ID of the planner override to retrieve.

Return type *canvasapi.planner.PlannerOverride*

get_planner_overrides(***kwargs*)
Retrieve a planner override for the current user

Calls GET /api/v1/planner/overrides

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.planner.PlannerOverride*

get_poll(*poll*, ***kwargs*)
Get a single poll, based on the poll id.

Calls GET /api/v1/polls/:id

Parameters **poll** (*int*) – The ID of the poll or the poll to change.

Return type *canvasapi.poll.Poll*

get_polls(***kwargs*)
Returns a paginated list of polls for the current user

Calls GET /api/1/polls

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.poll.Poll*

get_progress(*progress*, ***kwargs*)
Get a specific progress.

Calls GET /api/v1/progress/:id

Parameters **progress** (int, str or *canvasapi.progress.Progress*) – The object or ID of the progress to retrieve.

Return type *canvasapi.progress.Progress*

get_root_outcome_group()
Redirect to root outcome group for context

Calls GET /api/v1/global/root_outcome_group

Returns The OutcomeGroup of the context.

Return type *canvasapi.outcome.OutcomeGroup*

get_section(*section*, *use_sis_id=False*, ***kwargs*)
Get details about a specific section.

Calls GET /api/v1/sections/:id

Parameters

- **section** (*canvasapi.section.Section* or int) – The object or ID of the section to get.
- **use_sis_id** (*bool*) – Whether or not section_id is an sis ID. Defaults to *False*.

Return type *canvasapi.section.Section*

get_todo_items()

Return the current user's list of todo items, as seen on the user dashboard.

Calls GET /api/v1/users/self/todo

Return type dict

get_upcoming_events()

Return the current user's upcoming events, i.e. the same things shown in the dashboard 'Coming Up' sidebar.

Calls GET /api/v1/users/self/upcoming_events

Return type dict

get_user(*user*, *id_type=None*)

Retrieve a user by their ID. *id_type* denotes which endpoint to try as there are several different IDs that can pull the same user record from Canvas.

Refer to API documentation's User example to see the ID types a user can be retrieved with.

Calls GET /api/v1/users/:id

Parameters

- **user** (*canvasapi.user.User* or int) – The user's object or ID.
- **id_type** (*str*) – The ID type.

Return type *canvasapi.user.User*

get_user_participants(*appointment_group*, ***kwargs*)

List user participants in this appointment group.

Calls GET /api/v1/appointment_groups/:id/users

Parameters **appointment_group** (*canvasapi.appointment_group.AppointmentGroup* or int) – The object or ID of the appointment group.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.user.User*

list_appointment_groups(***kwargs*)

List appointment groups.

> **Warning:** Deprecated since version 0.10.0: Use canvasapi.canvas.Canvas.get_appointment_groups() instead.

Calls GET /api/v1/appointment_groups

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.appointment_group.AppointmentGroup*

list_calendar_events(***kwargs*)
List calendar events.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.canvas.Canvas.get_calendar_events()` instead.

Calls GET /api/v1/calendar_events

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.calendar_event.CalendarEvent*

list_group_participants(*appointment_group*, ***kwargs*)
List student group participants in this appointment group.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi. canvas.Canvas.get_group_participants()` instead.

Calls GET /api/v1/appointment_groups/:id/groups

Parameters **appointment_group** (*canvasapi.appointment_group.AppointmentGroup* or int) – The object or ID of the appointment group.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.Group*

list_user_participants(*appointment_group*, ***kwargs*)
List user participants in this appointment group.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi. canvas.Canvas.get_user_participants()` instead.

Calls GET /api/v1/appointment_groups/:id/users

Parameters **appointment_group** (*canvasapi.appointment_group.AppointmentGroup* or int) – The object or ID of the appointment group.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.user.User*

reserve_time_slot(*calendar_event*, *participant_id=None*, ***kwargs*)
Return single Calendar Event by id

Calls POST /api/v1/calendar_events/:id/reservations

Parameters

- **calendar_event** (*canvasapi.calendar_event.CalendarEvent* or int) – The object or ID of the calendar event.
- **participant_id** (*str*) – The ID of the participant, if given.

Return type *canvasapi.calendar_event.CalendarEvent*

search_accounts(***kwargs*)
Return a list of up to 5 matching account domains. Partial matches on name and domain are supported.

Calls GET /api/v1/accounts/search

Return type dict

search_all_courses(***kwargs*)

List all the courses visible in the public index. Returns a list of dicts, each containing a single course.

Calls GET /api/v1/search/all_courses

Return type *list*

search_recipients(***kwargs*)

Find valid recipients (users, courses and groups) that the current user can send messages to. Returns a list of mixed data types.

Calls GET /api/v1/search/recipients

Return type *list*

set_course_nickname(*course*, *nickname*)

Set a nickname for the given course. This will replace the course's name in the output of subsequent API calls, as well as in selected places in the Canvas web user interface.

Calls PUT /api/v1/users/self/course_nicknames/:course_id

Parameters

- **course** (`canvasapi.course.Course` or int) – The ID of the course.
- **nickname** (`str`) – The nickname for the course.

Return type `canvasapi.course.CourseNickname`

6.2 Account

class canvasapi.account.**Account**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (`dict`) – The JSON object to build this object with.

abort_sis_imports_pending(***kwargs*)

Aborts all pending (created, but not processed or processing) SIS imports for the current account.

Calls PUT /api/v1/accounts/:account_id/sis_imports/abort_all_pending

Returns True if the API responds with aborted=True, False otherwise.

Return type bool

activate_role(*role*, ***kwargs*)

Reactivate an inactive role.

Calls POST /api/v1/accounts/:account_id/roles/:id/activate

Parameters **role** (`canvasapi.account.Role` or int) – The object or ID of the role.

Return type `canvasapi.account.Role`

add_authentication_providers(***kwargs*)

Add external authentication providers for the account

Calls POST /api/v1/accounts/:account_id/authentication_providers

Return type *canvasapi.authentication_provider.AuthenticationProvider*

add_grading_standards (*title*, *grading_scheme_entry*, ***kwargs*)
Create a new grading standard for the account.

Calls POST /api/v1/accounts/:account_id/grading_standards

Parameters

- **title** (*str*) – The title for the Grading Standard
- **grading_scheme** (*list[dict]*) – A list of dictionaries containing keys for "name" and "value"

Return type canvasapi.grading_standards.GradingStandard

close_notification_for_user (*user*, *notification*)
If the user no long wants to see a notification, it can be excused with this call.

Calls DELETE /api/v1/accounts/:account_id/users/:user_id/account_notifications/:id

Parameters

- **user** (*canvasapi.user.User* or int) – The user object or ID to close the notificaton for.
- **notification** (*canvasapi.account.AccountNotification* or int) – The notification object or ID to close.

Return type *canvasapi.account.AccountNotification*

create_account (***kwargs*)
Create a new root account.

Calls POST /api/v1/accounts/:account_id/root_accounts

Return type *canvasapi.account.Account*

create_admin (*user*, ***kwargs*)
Flag an existing user as an admin of the current account.

Calls POST /api/v1/accounts/:account_id/admins

Parameters **user** (*canvasapi.user.User* or int) – The user object or ID to promote to admin.

Return type canvasapi.account.Admin

create_content_migration (*migration_type*, ***kwargs*)
Create a content migration.

Calls POST /api/v1/accounts/:account_id/content_migrations

Parameters **migration_type** (str or *canvasapi.content_migration.Migrator*) – The migrator type to use in this migration

Return type *canvasapi.content_migration.ContentMigration*

create_course (***kwargs*)
Create a course.

Calls POST /api/v1/accounts/:account_id/courses

Return type *canvasapi.course.Course*

create_enrollment_term(***kwargs*)
Create an enrollment term.

Calls POST /api/v1/accounts/:account_id/terms

Return type *canvasapi.enrollment_term.EnrollmentTerm*

create_external_tool(*name, privacy_level, consumer_key, shared_secret, **kwargs*)
Create an external tool in the current account.

Calls POST /api/v1/accounts/:account_id/external_tools

Parameters

- **name** (*str*) – The name of the tool
- **privacy_level** (*str*) – What information to send to the external tool. Options are "anonymous", "name_only", "public"
- **consumer_key** (*str*) – The consumer key for the external tool
- **shared_secret** (*str*) – The shared secret with the external tool

Return type *canvasapi.external_tool.ExternalTool*

create_group_category(*name, **kwargs*)
Create a Group Category

Calls POST /api/v1/accounts/:account_id/group_categories

Parameters **name** (*str*) – Name of group category.

Return type *canvasapi.group.GroupCategory*

create_notification(*account_notification, **kwargs*)
Create and return a new global notification for an account.

Calls POST /api/v1/accounts/:account_id/account_notifications

Parameters **account_notification** (*dict*) – The notification to create.

Return type *canvasapi.account.AccountNotification*

create_role(*label, **kwargs*)
Create a new course-level or account-level role.

Calls POST /api/v1/accounts/:account_id/roles

Parameters **label** (*str*) – The label for the role.

Return type *canvasapi.account.Role*

create_sis_import(*attachment, **kwargs*)
Create a new SIS import for the current account.

Calls POST /api/v1/accounts/:account_id/sis_imports

Parameters **attachment** (*file or str*) – A file handler or path of the file to import.

Return type canvasapi.sis_import.SisImport

create_subaccount(*account, **kwargs*)
Add a new sub-account to a given account.

Calls POST /api/v1/accounts/:account_id/sub_accounts

Parameters **account** (*str*) – The name of the account

Return type *canvasapi.account.Account*

create_user(*pseudonym*, ***kwargs*)

Create and return a new user and pseudonym for an account.

Calls POST /api/v1/accounts/:account_id/users

Parameters **pseudonym** (*dict*) – The pseudonym of the account.

Return type *canvasapi.user.User*

create_user_login(*user*, *login*, ***kwargs*)

Create a new login for an existing user in the given account

Calls POST /api/v1/accounts/:account_id/logins

Parameters

- **user** (*dict*) – The attributes of the user to create a login for
- **login** (*dict*) – The attributes of the login to create

Return type *canvasapi.login.Login*

deactivate_role(*role*, ***kwargs*)

Deactivate a custom role.

Calls DELETE /api/v1/accounts/:account_id/roles/:id

Parameters **role** (*canvasapi.account.Role* or int) – The object or ID of the role.

Return type *canvasapi.account.Role*

delete()

Delete the current account

Note: Cannot delete an account with active courses or active sub accounts. Cannot delete a root account.

Calls DELETE /api/v1/accounts/:account_id/sub_accounts/:id

Returns True if successfully deleted; False otherwise.

Return type bool

delete_grading_period(*grading_period*)

Delete a grading period for an account.

Calls DELETE /api/v1/accounts/:account_id/grading_periods/:id

Parameters **grading_period** (*canvasapi.grading_period.GradingPeriod* or int) – The GradingPeriod object or ID to delete.

Returns True if the grading period was deleted, False otherwise.

Return type bool

delete_user(*user*)

Delete a user record from a Canvas root account.

If a user is associated with multiple root accounts (in a multi-tenant instance of Canvas), this action will NOT remove them from the other accounts.

WARNING: This API will allow a user to remove themselves from the account. If they do this, they won't be able to make API calls or log into Canvas at that account.

Calls DELETE /api/v1/accounts/:account_id/users/:user_id

Parameters **user** (*canvasapi.user.User* or int) – The user object or ID to delete.

Return type *canvasapi.user.User*

get_admins(***kwargs*)
Get the paginated list of admins for the current account.

Calls GET /api/v1/accounts/:account_id/admins

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.account.Admin`

get_all_outcome_links_in_context()
Get all outcome links for context - BETA

Calls GET /api/v1/accounts/:account_id/outcome_group_links

Returns Paginated List of OutcomesLinks in the context.

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.outcome.OutcomeLink`*

get_authentication_provider(*authentication_provider*, ***kwargs*)
Get the specified authentication provider

Calls GET /api/v1/accounts/:account_id/authentication_providers/:id

Parameters **authentication_provider** (*`canvasapi.authentication_provider.AuthenticationProvider`* or int) – The object or ID of the authentication provider

Return type *`canvasapi.authentication_provider.AuthenticationProvider`*

get_authentication_providers(***kwargs*)
Return the list of authentication providers

Calls GET /api/v1/accounts/:account_id/authentication_providers

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.authentication_provider.AuthenticationProvider`*

get_content_migration(*content_migration*, ***kwargs*)
Retrive a content migration by its ID

Calls GET /api/v1/accounts/:account_id/content_migrations/:id

Parameters **content_migration** – The object or ID of the content migration to retrieve.

Return type *`canvasapi.content_migration.ContentMigration`*

get_content_migrations(***kwargs*)
List content migrations that the current account can view or manage.

Calls GET /api/v1/accounts/:account_id/content_migrations/

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.content_migration.ContentMigration`*

get_courses(***kwargs*)
Retrieve the list of courses in this account.

Calls GET /api/v1/accounts/:account_id/courses

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.course.Course`*

get_department_level_grade_data_completed()
Return the distribution of all concluded grades in the default term

Calls GET /api/v1/accounts/:account_id/analytics/completed/grades

Return type dict

`get_department_level_grade_data_current()`

Return the distribution of all available grades in the default term

Calls GET /api/v1/accounts/:account_id/analytics/current/grades

Return type dict

`get_department_level_grade_data_with_given_term(term_id)`

Return the distribution of all available or concluded grades with the given term

Calls GET /api/v1/accounts/:account_id/analytics/terms/:term_id/grades

Parameters **term_id** (*int or str*) – The ID of the term, or the strings "current" or "completed"

Return type dict

`get_department_level_participation_data_completed()`

Return page view hits all concluded courses in the default term

Calls GET /api/v1/accounts/:account_id/analytics/completed/activity

Return type dict

`get_department_level_participation_data_current()`

Return page view hits all available courses in the default term

Calls GET /api/v1/accounts/:account_id/analytics/current/activity

Return type dict

`get_department_level_participation_data_with_given_term(term_id)`

Return page view hits all available or concluded courses in the given term

Calls GET /api/v1/accounts/:account_id/analytics/terms/:term_id/activity

Parameters **term_id** (*int or str*) – The ID of the term, or the strings "current" or "completed"

Return type dict

`get_department_level_statistics_completed()`

Return all available numeric statistics about the department in the default term

Calls GET /api/v1/accounts/:account_id/analytics/current/statistics

Return type dict

`get_department_level_statistics_current()`

Return all available numeric statistics about the department in the default term

Calls GET /api/v1/accounts/:account_id/analytics/current/statistics

Return type dict

`get_department_level_statistics_with_given_term(term_id)`

Return numeric statistics about the department with the given term

Calls GET /api/v1/accounts/:account_id/analytics/terms/:term_id/statistics

Parameters **term_id** (*int or str*) – The ID of the term, or the strings "current" or "completed"

Return type dict

get_enrollment(*enrollment*, ***kwargs*)
Get an enrollment object by ID.

Calls GET /api/v1/accounts/:account_id/enrollments/:id

Parameters **enrollment** (*canvasapi.enrollment.Enrollment* or int) – The object or ID of the enrollment to retrieve.

Return type *canvasapi.enrollment.Enrollment*

get_enrollment_terms(***kwargs*)
List enrollment terms for a context.

Calls GET /api/v1/accounts/:account_id/terms

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.enrollment_term.EnrollmentTerm*

get_external_tool(*tool*)

Calls GET /api/v1/accounts/:account_id/external_tools/:external_tool_id

Parameters **tool** (*canvasapi.external_tool.ExternalTool* or int) – The object or ID of the tool

Return type *canvasapi.external_tool.ExternalTool*

get_external_tools(***kwargs*)

Calls GET /api/v1/accounts/:account_id/external_tools

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.external_tool.ExternalTool*

get_grading_periods(***kwargs*)
Return a list of grading periods for the associated account.

Calls GET /api/v1/accounts/:account_id/grading_periods

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.grading_period.GradingPeriod*

get_grading_standards(***kwargs*)
Get a PaginatedList of the grading standards available for the account.

Calls GET /api/v1/accounts/:account_id/grading_standards

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.grading_standards.GradingStandard`

get_group_categories(***kwargs*)
List group categories for a context.

Calls GET /api/v1/accounts/:account_id/group_categories

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.GroupCategory*

get_groups(***kwargs*)
Return a list of active groups for the specified account.

Calls GET /api/v1/accounts/:account_id/groups

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.Group*

get_index_of_reports(*report_type*)

Retrieve all reports that have been run for the account of a specific type.

Calls GET /api/v1/accounts/:account_id/reports/:report

Parameters **report_type** (*str*) – The type of report.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.account.AccountReport*

get_migration_systems(***kwargs*)

Return a list of migration systems.

Calls GET /api/v1/accounts/:account_id/content_migrations/migrators

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.content_migration.Migrator*

get_outcome_group(*group*)

Returns the details of the Outcome Group with the given id.

Calls GET /api/v1/accounts/:account_id/outcome_groups/:id

Parameters **group** (*canvasapi.outcome.OutcomeGroup* or int) – The outcome group object or ID to return.

Returns An outcome group object.

Return type *canvasapi.outcome.OutcomeGroup*

get_outcome_groups_in_context()

Get all outcome groups for context - BETA

Calls GET /api/v1/accounts/:account_id/outcome_groups

Returns Paginated List of OutcomesGroups in the context.

Return type canvasapi.paginated_list.PaginatedList of canvasapi.outcome.OutcomeGroups

get_outcome_import_status(*outcome_import*, ***kwargs*)

Get the status of an already created Outcome import. Pass 'latest' for the outcome import id for the latest import.

Calls GET /api/v1/accounts/:account_id/outcome_imports/:id

Parameters **outcome_import** (*canvasapi.outcome_import.OutcomeImport*, int, or string: "latest") – The outcome import object or ID to get the status of.

Return type *canvasapi.outcome_import.OutcomeImport*

get_reports()

Return a list of reports for the current context.

Calls GET /api/v1/accounts/:account_id/reports

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.account.AccountReport*

get_role(*role*)

Retrieve a role by ID.

Calls GET /api/v1/accounts/:account_id/roles/:id

Parameters **role** (*canvasapi.account.Role* or int) – The object or ID of the role.

Return type *canvasapi.account.Role*

get_roles(***kwargs*)
List the roles available to an account.

Calls GET /api/v1/accounts/:account_id/roles

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.account.Role*

get_root_outcome_group()
Redirect to root outcome group for context

Calls GET /api/v1/accounts/:account_id/root_outcome_group

Returns The OutcomeGroup of the context.

Return type *canvasapi.outcome.OutcomeGroup*

get_rubric(*rubric_id*, ***kwargs*)
Get a single rubric, based on rubric id.

Calls GET /api/v1/accounts/:account_id/rubrics/:id

Parameters **rubric_id** (*int*) – The ID of the rubric.

Return type *canvasapi.rubric.Rubric*

get_rubrics(***kwargs*)
Get the paginated list of active rubrics for the current account.

Calls GET /api/v1/accounts/:account_id/rubrics

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.rubric.Rubric*

get_single_grading_standard(*grading_standard_id*, ***kwargs*)
Get a single grading standard from the account.

Calls GET /api/v1/accounts/:account_id/grading_standards/:grading_standard_id

Parameters **grading_standard_id** (*int*) – The grading standard id

Return type `canvasapi.grading_standards.GradingStandard`

get_sis_import(*sis_import*, ***kwargs*)
Retrieve information on an individual SIS import from this account.

Calls GET /api/v1/accounts/:account_id/sis_imports/:id

Parameters **sis_import** (int, str or `canvasapi.sis_import.SisImport`) – The object or ID of the sis_import to retrieve.

Return type `canvasapi.sis_import.SisImport`

get_sis_imports(***kwargs*)
Get the paginated list of SIS imports for the current account.

Calls GET /api/v1/accounts/:account_id/sis_imports

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.sis_import.SisImport`

get_sis_imports_running(***kwargs*)
Get the paginated list of running SIS imports for the current account.

Calls GET /api/v1/accounts/:account_id/sis_imports/importing

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.sis_import.SisImport`

get_subaccounts(*recursive=False*)

List accounts that are sub-accounts of the given account.

Calls GET /api/v1/accounts/:account_id/sub_accounts

Parameters **recursive** (*bool*) – If true, the entire account tree underneath this account will be returned. If false, only direct sub-accounts of this account will be returned.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.account.Account*

get_user_logins(***kwargs*)

Given a user ID, return that user's logins for the given account.

Calls GET /api/v1/accounts/:account_id/logins

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.login.Login*

get_user_notifications(*user*)

Return a list of all global notifications in the account for this user. Any notifications that have been closed by the user will not be returned.

Calls GET /api/v1/accounts/:account_id/users/:user_id/account_notifications

Parameters **user** (*canvasapi.user.User* or int) – The user object or ID to retrieve notifications for.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.account.AccountNotification*

get_users(***kwargs*)

Retrieve a list of users associated with this account.

Calls GET /api/v1/accounts/:account_id/users

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.user.User*

import_outcome(*attachment*, ***kwargs*)

Import outcome into canvas.

Calls POST /api/v1/accounts/:account_id/outcome_imports

Parameters **attachment** (*file or str*) – A file handler or path of the file to import.

Return type *canvasapi.outcome_import.OutcomeImport*

list_authentication_providers(***kwargs*)

Return the list of authentication providers

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_authentication_providers()* instead.

Calls GET /api/v1/accounts/:account_id/authentication_providers

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.authentication_provider.AuthenticationProvider*

list_enrollment_terms(***kwargs*)

List enrollment terms for a context.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_enrollment_terms()* instead.

Calls GET /api/v1/accounts/:account_id/terms

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.enrollment_term.EnrollmentTerm*

list_group_categories(***kwargs*)

List group categories for a context.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_group_categories()* instead.

Calls GET /api/v1/accounts/:account_id/group_categories

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.group.GroupCategory*

list_groups(***kwargs*)

Return a list of active groups for the specified account.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_groups()* instead.

Calls GET /api/v1/accounts/:account_id/groups

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.group.Group*

list_roles(***kwargs*)

List the roles available to an account.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_roles()* instead.

Calls GET /api/v1/accounts/:account_id/roles

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.account.Role*

list_rubrics(***kwargs*)

Get the paginated list of active rubrics for the current account.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.account.Account.get_rubrics()* instead.

Calls GET /api/v1/accounts/:account_id/rubrics

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.rubric.Rubric`*

`list_user_logins`(***kwargs*)

Given a user ID, return that user's logins for the given account.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.account.Account.get_user_logins()`* instead.

Calls GET /api/v1/accounts/:account_id/logins

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.login.Login`*

`show_account_auth_settings`(***kwargs*)

Return the current state of each account level setting

Calls GET /api/v1/accounts/:account_id/sso_settings

Return type *`canvasapi.account.SSOSettings`*

`update`(***kwargs*)

Update an existing account.

Calls PUT /api/v1/accounts/:id

Returns True if the account was updated, False otherwise.

Return type bool

`update_account_auth_settings`(***kwargs*)

Return the current state of account level after updated

Calls PUT /api/v1/accounts/:account_id/sso_settings

Return type *`canvasapi.account.SSOSettings`*

`update_role`(*role*, ***kwargs*)

Update permissions for an existing role.

Calls PUT /api/v1/accounts/:account_id/roles/:id

Parameters **`role`** (*`canvasapi.account.Role`* or int) – The object or ID of the role.

Return type *`canvasapi.account.Role`*

6.3 AccountNotification

class `canvasapi.account.`**`AccountNotification`**(*requester*, *attributes*)

Parameters

- **`requester`** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **`attributes`** (*`dict`*) – The JSON object to build this object with.

6.4 AccountReport

class canvasapi.account.**AccountReport**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.5 Role

class canvasapi.account.**Role**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.6 SSOSettings

class canvasapi.account.**SSOSettings**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.7 AppointmentGroup

class canvasapi.appointment_group.**AppointmentGroup**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete this appointment group.

Calls DELETE /api/v1/appointment_groups/:id

Return type *canvasapi.appointment_group.AppointmentGroup*

edit(*appointment_group*, ***kwargs*)

Modify this appointment group.

Calls PUT /api/v1/appointment_groups/:id

Parameters **appointment_group** (*dict*) – Dict containing an array of context codes

Return type *canvasapi.appointment_group.AppointmentGroup*

6.8 Assignment

class canvasapi.assignment.**Assignment**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_override(***kwargs*)

Create an override for this assignment.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/overrides

Return type canvasapi.assignment.AssignmentOverride

delete(***kwargs*)

Delete this assignment.

Calls DELETE /api/v1/courses/:course_id/assignments/:id

Return type *canvasapi.assignment.Assignment*

edit(***kwargs*)

Modify this assignment.

Calls PUT /api/v1/courses/:course_id/assignments/:id

Return type *canvasapi.assignment.Assignment*

get_gradeable_students(***kwargs*)

List students eligible to submit the assignment.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/gradeable_students

Return type canvasapi.paginated_list.PaginatedList of canvasapi.user.UserDisplay

get_override(*override*, ***kwargs*)

Get a single assignment override with the given override id.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/overrides/:id

Parameters **override** (canvasapi.assignment.AssignmentOverride or int) – The object or ID of the override to get

Return type canvasapi.assignment.AssignmentOverride

get_overrides(***kwargs*)

Get a paginated list of overrides for this assignment that target sections/groups/students visible to the current user.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/overrides

Return type canvasapi.paginated_list.PaginatedList of canvasapi.assignment.AssignmentOverride

get_peer_reviews(***kwargs*)

Get a list of all Peer Reviews for this assignment.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/peer_reviews

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.peer_review.PeerReview`

`get_submission`(*user*, ***kwargs*)

Get a single submission, based on user id.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id

Parameters **`user`** (*`canvasapi.user.User`* or int) – The object or ID of the related user

Return type *`canvasapi.submission.Submission`*

`get_submissions`(***kwargs*)

Get all existing submissions for this assignment.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/submissions

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.submission.Submission`*

`submissions_bulk_update`(***kwargs*)

Update the grading and comments on multiple student's assignment submissions in an asynchronous job.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/ submissions/update_grades

Return type *`canvasapi.progress.Progress`*

`submit`(*submission*, *file=None*, ***kwargs*)

Makes a submission for an assignment.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/submissions

Parameters

- **`submission`** (*`dict`*) – The attributes of the submission.
- **`file`** (*`file or str`*) – A file to upload with the submission. (Optional, defaults to *None*. Submission type must be *online_upload*)

Return type *`canvasapi.submission.Submission`*

`upload_to_submission`(*file*, *user=u'self'*, ***kwargs*)

Upload a file to a submission.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/ submissions/:user_id/files

Parameters

- **`file`** (*`file or str`*) – The file or path of the file to upload.
- **`user`** (*`canvasapi.user.User`*, int, or str) – The object or ID of the related user, or 'self' for the current user. Defaults to 'self'.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.9 AssignmentGroup

class `canvasapi.assignment.`**`AssignmentGroup`**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete this assignment.

Calls DELETE /api/v1/courses/:course_id/assignment_groups/:assignment_group_id

Return type *canvasapi.assignment.AssignmentGroup*

edit(***kwargs*)

Modify this assignment group.

Calls PUT /api/v1/courses/:course_id/assignment_groups/:assignment_group_id

Return type *canvasapi.assignment.AssignmentGroup*

6.10 AuthenticationProvider

class canvasapi.authentication_provider.**AuthenticationProvider**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Delete the config

Calls DELETE /api/v1/accounts/:account_id/authentication_providers/:id

Return type *canvasapi.authentication_provider.AuthenticationProvider*

update(***kwargs*)

Update an authentication provider using the same options as the create endpoint

Calls PUT /api/v1/accounts/:account_id/authentication_providers/:id

Return type *canvasapi.authentication_provider.AuthenticationProvider*

6.11 Avatar

class canvasapi.avatar.**Avatar**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.12 BlueprintMigration

class canvasapi.blueprint.**BlueprintMigration**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

get_details(***kwargs*)

Return the changes that were made in a blueprint migration.

Calls GET /api/v1/courses/:course_id/blueprint_templates/:template_id /migrations/:id/details

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.blueprint.ChangeRecord*

get_import_details(***kwargs*)

Return changes that were made to a course with a blueprint.

Calls GET /api/v1/courses/:course_id/blueprint_subscriptions/ :subscription_id/migrations/:id/details

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.blueprint.ChangeRecord*

6.13 BlueprintSubscription

class canvasapi.blueprint.**BlueprintSubscription**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

list_blueprint_imports(***kwargs*)

Return a list of migrations imported into a course associated with a blueprint.

Calls GET /api/v1/courses/:course_id/blueprint_subscriptions/:subscription_id/ migrations

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.blueprint.BlueprintMigration*

show_blueprint_import(*migration*, ***kwargs*)

Return the status of an import into a course associated with a blueprint.

Calls GET /api/v1/courses/:course_id/blueprint_subscriptions/:subscription_id/ migrations/:id

Parameters **migration** (int or *canvasapi.blueprint.BlueprintMigration*) – migration id or object

Return type *canvasapi.blueprint.BlueprintMigration*

6.14 BlueprintTemplate

class `canvasapi.blueprint.`**`BlueprintTemplate`**(*requester*, *attributes*)

Parameters

- **`requester`** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **`attributes`** (*`dict`*) – The JSON object to build this object with.

`associated_course_migration`(***kwargs*)

Start a migration to update content in all associated courses.

Calls POST /api/v1/courses/:course_id/blueprint_templates/:template_id/migrations

Return type *`canvasapi.blueprint.BlueprintMigration`*

`change_blueprint_restrictions`(*content_type*, *content_id*, *restricted*, ***kwargs*)

Set or remove restrictions on a blueprint course object. Must have all three parameters for this function call to work.

Calls PUT /api/v1/courses/:course_id/blueprint_templates/:template_id/restrict_item

Parameters

- **`content_type`** (*`str`*) – type of object
- **`content_id`** (*`int`*) – id of the object
- **`restricted`** (*`bool`*) – whether it's restricted or not

Returns True if the restriction was succesfully applied.

Return type bool

`get_associated_courses`(***kwargs*)

Return a list of courses associated with the given blueprint.

Calls GET /api/v1/courses/:course_id/blueprint_templates/:template_id/ associated_courses

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.course.Course`*

`get_unsynced_changes`(***kwargs*)

Return changes made to associated courses of a blueprint course.

Calls GET /api/v1/courses/:course_id/blueprint_templates/:template_id/unsynced_changes

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.blueprint.ChangeRecord`*

`list_blueprint_migrations`(***kwargs*)

Return a paginated list of migrations for the template.

Calls GET api/v1/courses/:course_id/blueprint_templates/:template_id/migrations

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.blueprint.BlueprintMigration`*

`show_blueprint_migration`(*migration*, ***kwargs*)

Return the status of a blueprint migration.

Calls GET /api/v1/courses/:course_id/blueprint_templates/:template_id /migrations/:id

Parameters **migration** (int or *canvasapi.blueprint.BlueprintMigration*) – migration id or object

Return type *canvasapi.blueprint.BlueprintMigration*

update_associated_courses(***kwargs*)

Add or remove new associations for the blueprint template.

Calls PUT /api/v1/courses/:course_id/blueprint_templates/:template_id/update_associations

Returns True if the course was added or removed, False otherwise.

Return type bool

6.15 ChangeRecord

class canvasapi.blueprint.**ChangeRecord**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.16 Bookmark

class canvasapi.bookmark.**Bookmark**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Delete this bookmark.

Calls DELETE /api/v1/users/self/bookmarks/:id

Return type *canvasapi.bookmark.Bookmark*

edit(***kwargs*)

Modify this bookmark.

Calls PUT /api/v1/users/self/bookmarks/:id

Return type *canvasapi.bookmark.Bookmark*

6.17 CalendarEvent

class canvasapi.calendar_event.**CalendarEvent**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.

- **attributes** (*dict*) – The JSON object to build this object with.

delete (***kwargs*)
Delete this calendar event.

Calls DELETE /api/v1/calendar_events/:id

Return type *canvasapi.calendar_event.CalendarEvent*

edit (***kwargs*)
Modify this calendar event.

Calls PUT /api/v1/calendar_events/:id

Return type *canvasapi.calendar_event.CalendarEvent*

6.18 CommunicationChannel

class canvasapi.communication_channel.**CommunicationChannel** (*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete ()
Delete the current communication_channel

Calls DELETE /api/v1/users/:user_id/communication_channels/:id

Returns True if successfully deleted; False otherwise.

Return type bool

get_preference (*notification*)
Fetch the preference for the given notification for the given communication channel.

Calls GET /api/v1/users/:user_id/communication_channels/ :communication_channel_id/notification_preferences/:notification

Parameters **notification** (*str*) – The name of the notification.

Return type canvasapi.notification_preference.NotificationPreference

get_preference_categories (***kwargs*)
Fetch all notification preference categories for the given communication channel.

Calls GET /api/v1/users/:user_id/communication_channels/ :communication_channel_id/notification_preference_categories

Return type *list*

get_preferences (***kwargs*)
Fetch all preferences for the given communication channel.

Calls GET /api/v1/users/:user_id/communication_channels/:communication_channel_id/ notification_preferences

Return type *list*

list_preference_categories(***kwargs*)

Fetch all notification preference categories for the given communication channel.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.communication_channel.CommunicationChannel.get_preference_categories()` instead.

Calls GET /api/v1/users/:user_id/communication_channels/ :communication_channel_id/notification_preference_categories

Return type *list*

list_preferences(***kwargs*)

Fetch all preferences for the given communication channel.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.communication_channel.CommunicationChannel.get_preferences()` instead.

Calls GET /api/v1/users/:user_id/communication_channels/:communication_channel_id/ notification_preferences

Return type *list*

update_multiple_preferences(*notification_preferences*, ***kwargs*)

Change preferences for multiple notifications based on the category for a single communication channel.

Calls PUT /api/v1/users/self/communication_channels/:communication_channel_id/ notification_preferences

Parameters **notification_preferences** – Dict that indicates the frequency for different notification types.

Return type `canvasapi.notification_preference.NotificationPreference`

update_preference(*notification*, *frequency*, ***kwargs*)

Update the preference for the given notification for the given communication channel.

Calls PUT /api/v1/users/self/communication_channels/:communication_channel_id/ notification_preferences/:notification

Parameters

- **notification** (*str*) – The name of the notification.
- **frequency** (*str Can be 'immediately', 'daily', 'weekly', or 'never'*) – The desired frequency for this notification.

Return type `canvasapi.notification_preference.NotificationPreference`

update_preferences_by_catagory(*category*, *frequency*, ***kwargs*)

Change preferences for multiple notifications based on the category for a single communication channel.

Calls PUT /api/v1/users/self/communication_channels/:communication_channel_id/ notification_preference_categories/:category

Parameters

- **category** (*str*) – The name of the category. Must be parameterized e.g. The category Course Content should be course_content
- **frequency** (*str Can be 'immediately', 'daily', 'weekly', or 'never'*) – The desired frequency for this notification.

Return type `canvasapi.notification_preference.NotificationPreference`

6.19 ContentExport

class `canvasapi.content_export.`**ContentExport**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.20 ContentMigration

class `canvasapi.content_migration.`**ContentMigration**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

get_migration_issue(*migration_issue*, ***kwargs*)

List a single issue for this content migration.

Calls GET /api/v1/accounts/:account_id/content_migrations/:content_migration_id/migration_issues/:id

or GET /api/v1/courses/:course_id/content_migrations/:content_migration_id/migration_issues/:id

or GET /api/v1/groups/:group_id/content_migrations/:content_migration_id/migration_issues/:id

or GET /api/v1/users/:user_id/content_migrations/:content_migration_id/migration_issues/:id

Parameters migration_issue (int, str or *canvasapi.content_migration.ContentMigration*) – The object or ID of the issue to retrieve.

Return type *canvasapi.content_migration.MigrationIssue*

get_migration_issues(***kwargs*)

List issues for this content migration

Calls GET /api/v1/accounts/:account_id/content_migrations/:content_migration_id/migration_issues

or GET /api/v1/courses/:course_id/content_migrations/:content_migration_id/migration_issues

or GET /api/v1/groups/:group_id/content_migrations/:content_migration_id/migration_issues

or GET /api/v1/users/:user_id/content_migrations/:content_migration_id/migration_issues

Return type *canvasapi.content_migration.MigrationIssue*

get_parent(***kwargs*)

Return the object that spawned this content migration.

Return type `canvasapi.group.Account`, or *canvasapi.course.Course*, or `canvasapi.course.Group`, or `canvasapi.course.User`

get_progress(***kwargs*)
Get the progress of the current content migration.

Calls GET /api/v1/progress/:id

Return type *canvasapi.progress.Progress*

update(***kwargs*)
Update an existing content migration.

Calls PUT /api/v1/accounts/:account_id/content_migrations/:id

or PUT /api/v1/courses/:course_id/content_migrations/:id

or PUT /api/v1/groups/:group_id/content_migrations/:id

or PUT /api/v1/users/:user_id/content_migrations/:id

Returns True if the migration was updated, False otherwise.

Return type bool

class canvasapi.content_migration.**MigrationIssue**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

update(***kwargs*)
Update an existing migration issue.

Calls PUT /api/v1/accounts/:account_id/content_migrations/:content_migration_id/migration_issues/:id
or PUT /api/v1/courses/:course_id/content_migrations/:content_migration_id/migration_issues/:id
or PUT /api/v1/groups/:group_id/content_migrations/:content_migration_id/migration_issues/:id
or PUT /api/v1/users/:user_id/content_migrations/:content_migration_id/migration_issues/:id

Returns True if the issue was updated, False otherwise.

Return type bool

class canvasapi.content_migration.**Migrator**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.21 Conversation

class canvasapi.conversation.**Conversation**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

add_message(*body*, ***kwargs*)
Add a message to a conversation.

Calls POST /api/v1/conversations/:id/add_message

Parameters **body** (`str`) – The body of the conversation.

Returns A conversation with only the most recent message.

Return type `canvasapi.account.Conversation`

add_recipients(*recipients*)
Add a recipient to a conversation.

Calls POST /api/v1/conversations/:id/add_recipients

Parameters **recipients** (*list* of *str*) – A list of string format recipient ids. These may be user ids or course/group ids prefixed with 'course_' or 'group_' respectively, e.g. recipients['1', '2', 'course_3']

Return type `canvasapi.account.Conversation`

delete()
Delete a conversation.

Calls DELETE /api/v1/conversations/:id

Return type *bool*

delete_messages(*remove*)
Delete messages from this conversation.

Note that this only affects this user's view of the conversation. If all messages are deleted, the conversation will be as well.

Calls POST /api/v1/conversations/:id/remove_messages

Parameters **remove** (*list* of *str*) – List of message ids to be removed.

Return type *dict*

edit(***kwargs*)
Update a conversation.

Calls PUT /api/v1/conversations/:id

Return type *bool*

6.22 Course

class canvasapi.course.**Course**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (`dict`) – The JSON object to build this object with.

add_grading_standards(*title*, *grading_scheme_entry*, ***kwargs*)
Create a new grading standard for the course.

Calls POST /api/v1/courses/:course_id/grading_standards

Parameters

- **title** (*str*) – The title for the Grading Standard
- **grading_scheme** (*list[dict]*) – A list of dictionaries containing keys for "name" and "value"

Return type canvasapi.grading_standards.GradingStandard

conclude()

Mark this course as concluded.

Calls DELETE /api/v1/courses/:id

Returns True if the course was concluded, False otherwise.

Return type bool

create_assignment(*assignment*, ***kwargs*)

Create a new assignment for this course.

Note: The assignment is created in the active state.

Calls POST /api/v1/courses/:course_id/assignments

Parameters **assignment** (*dict*) – The attributes of the assignment

Return type *canvasapi.assignment.Assignment*

create_assignment_group(***kwargs*)

Create a new assignment group for this course.

Calls POST /api/v1/courses/:course_id/assignment_groups

Return type *canvasapi.assignment.AssignmentGroup*

create_assignment_overrides(*assignment_overrides*, ***kwargs*)

Create the specified overrides for each assignment.

Calls POST /api/v1/courses/:course_id/assignments/overrides

Parameters **assignment_overrides** (*list*) – Attributes for the new assignment overrides.

Return type canvasapi.paginated_list.PaginatedList of canvasapi.assignment.AssignmentOverride

create_content_migration(*migration_type*, ***kwargs*)

Create a content migration.

Calls POST /api/v1/courses/:course_id/content_migrations

Parameters **migration_type** (str or *canvasapi.content_migration.Migrator*) – The migrator type to use in this migration

Return type *canvasapi.content_migration.ContentMigration*

create_course_section(***kwargs*)

Create a new section for this course.

Calls POST /api/v1/courses/:course_id/sections

Return type canvasapi.course.Section

create_discussion_topic(***kwargs*)

Creates a new discussion topic for the course or group.

Calls POST /api/v1/courses/:course_id/discussion_topics

Return type *canvasapi.discussion_topic.DiscussionTopic*

create_epub_export(***kwargs*)
Create an ePub export for a course.

Calls POST /api/v1/courses/:course_id/epub_exports/:id

Return type *canvasapi.course_epub_export.CourseEpubExport*

create_external_feed(*url*, ***kwargs*)
Create a new external feed for the course.

Calls POST /api/v1/courses/:course_id/external_feeds

Parameters **url** (*str*) – The url of the external rss or atom feed

Return type canvasapi.external_feed.ExternalFeed

create_external_tool(*name*, *privacy_level*, *consumer_key*, *shared_secret*, ***kwargs*)
Create an external tool in the current course.

Calls POST /api/v1/courses/:course_id/external_tools

Parameters

- **name** (*str*) – The name of the tool
- **privacy_level** (*str*) – What information to send to the external tool. Options are "anonymous", "name_only", "public"
- **consumer_key** (*str*) – The consumer key for the external tool
- **shared_secret** (*str*) – The shared secret with the external tool

Return type *canvasapi.external_tool.ExternalTool*

create_folder(*name*, ***kwargs*)
Creates a folder in this course.

Calls POST /api/v1/courses/:course_id/folders

Parameters **name** (*str*) – The name of the folder.

Return type *canvasapi.folder.Folder*

create_group_category(*name*, ***kwargs*)
Create a group category.

Calls POST /api/v1/courses/:course_id/group_categories

Parameters **name** (*str*) – Name of the category.

Return type *canvasapi.group.GroupCategory*

create_module(*module*, ***kwargs*)
Create a new module.

Calls POST /api/v1/courses/:course_id/modules

Parameters **module** (*dict*) – The attributes for the module.

Returns The created module.

Return type *canvasapi.module.Module*

create_page(*wiki_page*, ***kwargs*)
Create a new wiki page.

Calls POST /api/v1/courses/:course_id/pages

Parameters **title** (*dict*) – The title for the page.

Returns The created page.

Return type *canvasapi.course.Course*

create_quiz (*quiz*, ***kwargs*)
Create a new quiz in this course.

Calls POST /api/v1/courses/:course_id/quizzes

Parameters **quiz** (*dict*) – The attributes for the quiz.

Return type *canvasapi.quiz.Quiz*

delete ()
Permanently delete this course.

Calls DELETE /api/v1/courses/:id

Returns True if the course was deleted, False otherwise.

Return type bool

delete_external_feed (*feed*)
Deletes the external feed.

Calls DELETE /api/v1/courses/:course_id/external_feeds/:external_feed_id

Parameters **feed** (`canvasapi.external_feed.ExternalFeed` or int) – The object or ID of the feed to be deleted.

Return type `canvasapi.external_feed.ExternalFeed`

edit_front_page (***kwargs*)
Update the title or contents of the front page.

Calls PUT /api/v1/courses/:course_id/front_page

Return type *canvasapi.course.Course*

enroll_user (*user*, *enrollment_type*, ***kwargs*)
Create a new user enrollment for a course or a section.

Calls POST /api/v1/courses/:course_id/enrollments

Parameters

- **user** (*canvasapi.user.User* or int) – The object or ID of the user to enroll in this course.
- **enrollment_type** (*str*) – The type of enrollment.

Return type *canvasapi.enrollment.Enrollment*

export_content (*export_type*, ***kwargs*)
Begin a content export job for a course.

Calls POST /api/v1/courses/:course_id/content_exports

Parameters **export_type** (*str*) – The type of content to export.

Return type *canvasapi.content_export.ContentExport*

get_all_outcome_links_in_context ()
Get all outcome links for context - BETA

Calls GET /api/v1/courses/:course_id/outcome_group_links

Returns Paginated List of OutcomesLinks in the context.

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.outcome.OutcomeLink`*

`get_assignment` (*assignment*, ***kwargs*)
Return the assignment with the given ID.

Calls GET /api/v1/courses/:course_id/assignments/:id

Parameters **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the assignment to retrieve.

Return type *`canvasapi.assignment.Assignment`*

`get_assignment_group` (*assignment_group*, ***kwargs*)
Retrieve specified assignment group for the specified course.

Calls GET /api/v1/courses/:course_id/assignment_groups/:assignment_group_id

Parameters **`assignment_group`** (*`canvasapi.assignment.AssignmentGroup`* or int) – object or ID of assignment group.

Return type *`canvasapi.assignment.AssignmentGroup`*

`get_assignment_groups` (***kwargs*)
List assignment groups for the specified course.

Calls GET /api/v1/courses/:course_id/assignment_groups

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.assignment.AssignmentGroup`*

`get_assignment_overrides` (*assignment_overrides*, ***kwargs*)

List the specified overrides in this course, providing they target sections/groups/students visible to the current user.

Calls GET /api/v1/courses/:course_id/assignments/overrides

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.assignment.AssignmentOverride`

`get_assignments` (***kwargs*)
List all of the assignments in this course.

Calls GET /api/v1/courses/:course_id/assignments

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.assignment.Assignment`*

`get_blueprint` (*template=u'default'*, ***kwargs*)
Return the blueprint of a given ID.

Calls GET /api/v1/courses/:course_id/blueprint_templates/:template_id

Parameters **`template`** (int or *`canvasapi.blueprint.BlueprintTemplate`*) – The object or ID of the blueprint template to get.

Return type *`canvasapi.blueprint.BlueprintTemplate`*

`get_content_export` (*content_export*, ***kwargs*)
Return information about a single content export.

Calls GET /api/v1/courses/:course_id/content_exports/:id

Parameters **content_export** (int or *canvasapi.content_export.ContentExport*) – The object or ID of the content export to show.

Return type *canvasapi.content_export.ContentExport*

get_content_exports(***kwargs*)
Return a paginated list of the past and pending content export jobs for a course.

Calls GET /api/v1/courses/:course_id/content_exports

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.content_export.ContentExport*

get_content_migration(*content_migration*, ***kwargs*)
Retrive a content migration by its ID

Calls GET /api/v1/courses/:course_id/content_migrations/:id

Parameters **content_migration** (int, str or *canvasapi.content_migration.ContentMigration*) – The object or ID of the content migration to retrieve.

Return type *canvasapi.content_migration.ContentMigration*

get_content_migrations(***kwargs*)
List content migrations that the current account can view or manage.

Calls GET /api/v1/courses/:course_id/content_migrations/

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.content_migration.ContentMigration*

get_course_level_assignment_data(***kwargs*)
Return a list of assignments for the course sorted by due date

Calls GET /api/v1/courses/:course_id/analytics/assignments

Return type dict

get_course_level_participation_data()
Return page view hits and participation numbers grouped by day through the course's history

Calls GET /api/v1/courses/:course_id/analytics/activity

Return type dict

get_course_level_student_summary_data(***kwargs*)
Return a summary of per-user access information for all students in a course

Calls GET /api/v1/courses/:course_id/analytics/student_summaries

Return type dict

get_discussion_topic(*topic*)
Return data on an individual discussion topic.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id

Parameters **topic** (*canvasapi.discussion_topic.DiscussionTopic* or int) – The object or ID of the discussion topic.

Return type *canvasapi.discussion_topic.DiscussionTopic*

get_discussion_topics(***kwargs*)
Returns the paginated list of discussion topics for this course or group.

Calls GET /api/v1/courses/:course_id/discussion_topics

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.discussion_topic.DiscussionTopic`*

`get_enrollments`(***kwargs*)
List all of the enrollments in this course.

Calls GET /api/v1/courses/:course_id/enrollments

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.enrollment.Enrollment`*

`get_epub_export`(*epub*, ***kwargs*)
Get information about a single epub export.

Calls GET /api/v1/courses/:course_id/epub_exports/:id

Parameters **`epub`** (int or *`canvasapi.course_epub_export.CourseEpubExport`*) – Object or ID of ePub Export

Return type *`canvasapi.course_epub_export.CourseEpubExport`*

`get_external_feeds`(***kwargs*)
Returns the list of External Feeds this course.

Calls GET /api/v1/courses/:course_id/external_feeds

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.external_feed.ExternalFeed`

`get_external_tool`(*tool*)

Calls GET /api/v1/courses/:course_id/external_tools/:external_tool_id

Parameters **`tool`** (*`canvasapi.external_tool.ExternalTool`* or int) – The object or ID of the tool to retrieve.

Return type *`canvasapi.external_tool.ExternalTool`*

`get_external_tools`(***kwargs*)

Calls GET /api/v1/courses/:course_id/external_tools

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.external_tool.ExternalTool`*

`get_file`(*file*, ***kwargs*)
Return the standard attachment json object for a file.

Calls GET /api/v1/courses/:course_id/files/:id

Parameters **`file`** (*`canvasapi.file.File`* or int) – The object or ID of the file to retrieve.

Return type *`canvasapi.file.File`*

`get_files`(***kwargs*)
Returns the paginated list of files for the course.

Calls GET /api/v1/courses/:course_id/files

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.file.File`*

`get_folder`(*folder*)
Returns the details for a course folder

Calls GET /api/v1/courses/:course_id/folders/:id

Parameters **folder** (*canvasapi.folder.Folder* or int) – The object or ID of the folder to retrieve.

Return type *canvasapi.folder.Folder*

get_folders(***kwargs*)
Returns the paginated list of all folders for the given course. This will be returned as a flat list containing all subfolders as well.

Calls GET /api/v1/courses/:course_id/folders

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.folder.Folder*

get_full_discussion_topic(*topic*)
Return a cached structure of the discussion topic.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/view

Parameters **topic** (*canvasapi.discussion_topic.DiscussionTopic* or int) – The object or ID of the discussion topic.

Return type dict

get_grading_period(*grading_period*, ***kwargs*)
Return a single grading period for the associated course and id.

Calls GET /api/v1/courses/:course_id/grading_periods/:id

Parameters **grading_period_id** (*int*) – The ID of the rubric.

Return type *canvasapi.grading_period.GradingPeriod*

get_grading_periods(***kwargs*)
Return a list of grading periods for the associated course.

Calls GET /api/v1/courses/:course_id/grading_periods

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.grading_period.GradingPeriod*

get_grading_standards(***kwargs*)
Get a PaginatedList of the grading standards available for the course

Calls GET /api/v1/courses/:course_id/grading_standards

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.grading_standards.GradingStandard`

get_group_categories(***kwargs*)
List group categories for a context.

Calls GET /api/v1/courses/:course_id/group_categories

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.GroupCategory*

get_groups(***kwargs*)
Return list of active groups for the specified course.

Calls GET /api/v1/courses/:course_id/groups

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.course.Course*

get_migration_systems(***kwargs*)

Return a list of migration systems.

Calls GET /api/v1/courses/:course_id/content_migrations/migrators

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_migration.Migrator*

get_module(*module*, ***kwargs*)

Retrieve a single module by ID.

Calls GET /api/v1/courses/:course_id/modules/:id

Parameters **module** (*canvasapi.module.Module* or int) – The object or ID of the module to retrieve.

Return type *canvasapi.module.Module*

get_modules(***kwargs*)

Return a list of modules in this course.

Calls GET /api/v1/courses/:course_id/modules

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.module.Module*

get_multiple_submissions(***kwargs*)

List submissions for multiple assignments. Get all existing submissions for a given set of students and assignments.

Calls GET /api/v1/courses/:course_id/students/submissions

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.submission.Submission*

get_outcome_group(*group*)

Returns the details of the Outcome Group with the given id.

Calls GET /api/v1/courses/:course_id/outcome_groups/:id

Parameters **group** (*canvasapi.outcome.OutcomeGroup* or int) – The outcome group object or ID to return.

Returns An outcome group object.

Return type *canvasapi.outcome.OutcomeGroup*

get_outcome_groups_in_context()

Get all outcome groups for context - BETA

Calls GET /api/v1/courses/:course_id/outcome_groups

Returns Paginated List of OutcomesGroups in the context.

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.outcome.OutcomeGroups`

get_outcome_import_status(*outcome_import*, ***kwargs*)

Get the status of an already created Outcome import. Pass 'latest' for the outcome import id for the latest import.

Calls GET /api/v1/courses/:course_id/outcome_imports/:id

Parameters **outcome_import** (*canvasapi.outcome_import.OutcomeImport*, int, or string: "latest") – The outcome import object or ID to get the status of.

Return type *canvasapi.outcome_import.OutcomeImport*

get_outcome_result_rollups (***kwargs*)
Get all outcome result rollups for context - BETA

Calls GET /api/v1/courses/:course_id/outcome_rollups

Returns List of outcome result rollups in the context.

Return type dict

get_outcome_results (***kwargs*)
Get all outcome results for context - BETA

Calls GET /api/v1/courses/:course_id/outcome_results

Returns List of potential related outcome result dicts.

Return type dict

get_page (*url*)
Retrieve the contents of a wiki page.

Calls GET /api/v1/courses/:course_id/pages/:url

Parameters **url** (*str*) – The url for the page.

Returns The specified page.

Return type *canvasapi.course.Course*

get_pages (***kwargs*)
List the wiki pages associated with a course.

Calls GET /api/v1/courses/:course_id/pages

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.course.Course*

get_quiz (*quiz*)
Return the quiz with the given id.

Calls GET /api/v1/courses/:course_id/quizzes/:id

Parameters **quiz** (*canvasapi.quiz.Quiz* or int) – The object or ID of the quiz to retrieve.

Return type *canvasapi.quiz.Quiz*

get_quizzes (***kwargs*)
Return a list of quizzes belonging to this course.

Calls GET /api/v1/courses/:course_id/quizzes

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.quiz.Quiz*

get_recent_students ()
Return a list of students in the course ordered by how recently they have logged in.

Calls GET /api/v1/courses/:course_id/recent_students

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.user.User*

get_root_outcome_group ()
Redirect to root outcome group for context

Calls GET /api/v1/courses/:course_id/root_outcome_group

Returns The OutcomeGroup of the context.

Return type *canvasapi.outcome.OutcomeGroup*

get_rubric(*rubric_id*, ***kwargs*)
Get a single rubric, based on rubric id.

Calls GET /api/v1/courses/:course_id/rubrics/:id

Parameters **rubric_id** (*int*) – The ID of the rubric.

Return type *canvasapi.rubric.Rubric*

get_rubrics(***kwargs*)
Get the paginated list of active rubrics for the current course.

Calls GET /api/v1/courses/:course_id/rubrics

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.rubric.Rubric*

get_section(*section*, ***kwargs*)
Retrieve a section.

Calls GET /api/v1/courses/:course_id/sections/:id

Parameters **section** (*canvasapi.section.Section* or int) – The object or ID of the section to retrieve.

Return type *canvasapi.section.Section*

get_sections(***kwargs*)
List all sections in a course.

Calls GET /api/v1/courses/:course_id/sections

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.section.Section*

get_settings()
Returns this course’s settings.

Calls GET /api/v1/courses/:course_id/settings

Return type dict

get_single_grading_standard(*grading_standard_id*, ***kwargs*)
Get a single grading standard from the course.

Calls GET /api/v1/courses/:course_id/grading_standards/:grading_standard_id

Parameters **grading_standard_id** (*int*) – The grading standard id

Return type canvasapi.grading_standards.GradingStandard

get_submission(*assignment*, *user*, ***kwargs*)
Get a single submission, based on user id.

> **Warning:** Deprecated since version 0.9.0: Use *canvasapi.assignment.Assignment.get_submission()* instead.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id

Parameters

- **assignment** (*canvasapi.assignment.Assignment* or int) – The object or ID of the related assignment
- **user** (*canvasapi.user.User* or int) – The object or ID of the related user

Return type *canvasapi.submission.Submission*

get_tabs (***kwargs*)

List available tabs for a course. Returns a list of navigation tabs available in the current context.

Calls GET /api/v1/courses/:course_id/tabs

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.tab.Tab*

get_user (*user*, *user_id_type=None*)

Retrieve a user by their ID. *user_id_type* denotes which endpoint to try as there are several different ids that can pull the same user record from Canvas.

Calls GET /api/v1/courses/:course_id/users/:id

Parameters

- **user** (*canvasapi.user.User* or int) – The object or ID of the user to retrieve.
- **user_id_type** (*str*) – The type of the ID to search for.

Return type *canvasapi.user.User*

get_user_in_a_course_level_assignment_data (*user*)

Return a list of assignments for the course sorted by due date

Calls GET /api/v1/courses/:course_id/analytics/users/:student_id/assignments

Parameters **user** (*canvasapi.user.User* or int) – The object or ID of the related user

Return type dict

get_user_in_a_course_level_messaging_data (*user*)

Return messaging hits grouped by day through the entire history of the course

Calls GET /api/v1/courses/:course_id/analytics/users/:student_id/communication

Parameters **user** (*canvasapi.user.User* or int) – The object or ID of the related user

Return type dict

get_user_in_a_course_level_participation_data (*user*)

Return page view hits grouped by hour and participation details through course’s history

Calls GET /api/v1/courses/:course_id/analytics/users/:student_id/activity

Parameters **user** (*canvasapi.user.User* or int) – The object or ID of the related user

Return type dict

get_users (***kwargs*)

List all users in a course.

Calls GET /api/v1/courses/:course_id/search_users

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.user.User*

import_outcome (*attachment*, ***kwargs*)

Import outcome into canvas.

Calls POST /api/v1/courses/:course_id/outcome_imports

Parameters **attachment** (*file or str*) – A file handler or path of the file to import.

Return type *canvasapi.outcome_import.OutcomeImport*

list_assignment_groups(***kwargs*)

List assignment groups for the specified course.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.course.Course.get_assignment_groups()* instead.

Calls GET /api/v1/courses/:course_id/assignment_groups

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.assignment.AssignmentGroup*

list_blueprint_subscriptions(***kwargs*)

Return a list of blueprint subscriptions for the given course.

Calls GET /api/v1/courses/:course_id/blueprint_subscriptions

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.blueprint.BlueprintSubscription*

list_external_feeds(***kwargs*)

Returns the list of External Feeds this course.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.course.Course.get_external_feeds()* instead.

Calls GET /api/v1/courses/:course_id/external_feeds

Return type canvasapi.paginated_list.PaginatedList of canvasapi.external_feed.ExternalFeed

list_files(***kwargs*)

Returns the paginated list of files for the course.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.course.Course.get_files()* instead.

Calls GET /api/v1/courses/:course_id/files

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.file.File*

list_folders(***kwargs*)

Returns the paginated list of all folders for the given course. This will be returned as a flat list containing all subfolders as well.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.course.Course.get_folders()* instead.

Calls GET /api/v1/courses/:course_id/folders

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.folder.Folder`*

`list_gradeable_students`(*assignment*, ***kwargs*)

List students eligible to submit the assignment.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.assignment.Assignment.get_gradeable_students()`* instead.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/gradeable_students

Parameters **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the related assignment

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.user.UserDisplay`

`list_group_categories`(***kwargs*)

List group categories for a context.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.course.Course.get_group_categories()`* instead.

Calls GET /api/v1/courses/:course_id/group_categories

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.group.GroupCategory`*

`list_groups`(***kwargs*)

Return list of active groups for the specified course.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.course.Course.get_groups()`* instead.

Calls GET /api/v1/courses/:course_id/groups

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.course.Course`*

`list_multiple_submissions`(***kwargs*)

List submissions for multiple assignments. Get all existing submissions for a given set of students and assignments.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.course.Course.get_multiple_submissions()`* instead.

Calls GET /api/v1/courses/:course_id/students/submissions

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.submission.Submission`*

`list_rubrics`(***kwargs*)

Get the paginated list of active rubrics for the current course.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.course.Course.get_rubrics()`* instead.

Calls GET /api/v1/courses/:course_id/rubrics

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.rubric.Rubric`*

`list_sections`(***kwargs*)

Returns the list of sections for this course.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.course.Course.get_sections()`* instead.

Calls GET /api/v1/courses/:course_id/sections

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.section.Section`*

`list_submissions`(*assignment*, ***kwargs*)

Get all existing submissions for an assignment.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.assignment.Assignment.get_submissions()`* instead.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/submissions

Parameters **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the related assignment

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.submission.Submission`*

`list_tabs`(***kwargs*)

List available tabs for a course. Returns a list of navigation tabs available in the current context.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.course.Course.get_tabs()`* instead.

Calls GET /api/v1/courses/:course_id/tabs

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.tab.Tab`*

mark_submission_as_read(*assignment*, *user*, ***kwargs*)

Mark submission as read. No request fields are necessary.

> **Warning:** Deprecated since version 0.9.0: Use `canvasapi.submission.Submission.mark_read()` instead.

Calls PUT /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id/read

Parameters

- **assignment** (`canvasapi.assignment.Assignment` or int) – The object or ID of the related assignment
- **user** (`canvasapi.user.User` or int) – The object or ID of the related user

Return type *bool*

mark_submission_as_unread(*assignment*, *user*, ***kwargs*)

Mark submission as unread. No request fields are necessary.

> **Warning:** Deprecated since version 0.9.0: Use `canvasapi.submission.Submission.mark_unread()` instead.

Calls DELETE /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id/read

Parameters

- **assignment** (`canvasapi.assignment.Assignment` or int) – The object or ID of the related assignment
- **user** (`canvasapi.user.User` or int) – The object or ID of the related user

Return type *bool*

preview_html(*html*)

Preview HTML content processed for this course.

Calls POST /api/v1/courses/:course_id/preview_html

Parameters **html** (*str*) – The HTML code to preview.

Return type str

reorder_pinned_topics(*order*)

Puts the pinned discussion topics in the specified order. All pinned topics should be included.

Calls POST /api/v1/courses/:course_id/discussion_topics/reorder

Parameters **order** (*string or iterable sequence of values*) – The ids of the pinned discussion topics in the desired order. e.g. [104, 102, 103], (104, 102, 103), or "104,102,103"

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.discussion_topic.DiscussionTopic`

reset()

Delete the current course and create a new equivalent course with no content, but all sections and users moved over.

Calls POST /api/v1/courses/:course_id/reset_content

Return type *canvasapi.course.Course*

set_quiz_extensions(*quiz_extensions*, ***kwargs*)

Set extensions for student all quiz submissions in a course.

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/extensions

Parameters **quiz_extensions** (*list*) – List of dictionaries representing extensions.

Return type list of *canvasapi.quiz.QuizExtension*

Example Usage:

```
>>> course.set_quiz_extensions([
...     {
...         'user_id': 1,
...         'extra_time': 60,
...         'extra_attempts': 1
...     },
...     {
...         'user_id': 2,
...         'extra_attempts': 3
...     },
...     {
...         'user_id': 3,
...         'extra_time': 20
...     }
... ])
```

show_front_page()

Retrieve the content of the front page.

Calls GET /api/v1/courses/:course_id/front_page

Return type *canvasapi.course.Course*

submissions_bulk_update(***kwargs*)

Update the grading and comments on multiple student's assignment submissions in an asynchronous job.

Calls POST /api/v1/courses/:course_id/submissions/update_grades

Return type *canvasapi.progress.Progress*

submit_assignment(*assignment*, *submission*, ***kwargs*)

Makes a submission for an assignment.

> **Warning:** Deprecated since version 0.9.0: Use *canvasapi.assignment.Assignment.submit()* instead.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/submissions

Parameters

- **assignment** (*canvasapi.assignment.Assignment* or int) – The object or ID of the related assignment
- **submission** (*dict*) – The attributes of the submission.

Return type *canvasapi.submission.Submission*

update(***kwargs*)
Update this course.

Calls PUT /api/v1/courses/:id

Returns True if the course was updated, False otherwise.

Return type bool

update_assignment_overrides(*assignment_overrides*, ***kwargs*)
Update a list of specified overrides for each assignment.

Note: All current overridden values must be supplied if they are to be retained.

Calls PUT /api/v1/courses/:course_id/assignments/overrides

Parameters **assignment_overrides** (`list`) – Attributes for the updated assignment overrides.

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.assignment.AssignmentOverride`

update_settings(***kwargs*)
Update a course's settings.

Calls PUT /api/v1/courses/:course_id/settings

Return type dict

update_submission(*assignment*, *user*, ***kwargs*)
Comment on and/or update the grading for a student's assignment submission.

> **Warning:** Deprecated since version 0.9.0: Use `canvasapi.submission.Submission.edit()` instead.

Calls PUT /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id

Parameters

- **assignment** (`canvasapi.assignment.Assignment` or int) – The object or ID of the related assignment
- **user** (`canvasapi.user.User` or int) – The object or ID of the related user

Return type `canvasapi.submission.Submission`

update_tab(*tab_id*, ***kwargs*)
Update a tab for a course.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.tab.Tab.update()` instead.

Calls PUT /api/v1/courses/:course_id/tabs/:tab_id

Parameters **tab_id** (`str`) – The ID of the tab

Return type `canvasapi.tab.Tab`

upload(*file*, ***kwargs*)
Upload a file to this course.

Calls POST /api/v1/courses/:course_id/files

Parameters **file** (*file or str*) – The file or path of the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.23 CourseNickname

class canvasapi.course.**CourseNickname**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

remove()

Remove the nickname for the given course. Subsequent course API calls will return the actual name for the course.

Calls DELETE /api/v1/users/self/course_nicknames/:course_id

Return type *canvasapi.course.CourseNickname*

6.24 CourseEpubExport

class canvasapi.course_epub_export.**CourseEpubExport**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.25 CurrentUser

class canvasapi.current_user.**CurrentUser**(*_requester*)

add_favorite_course(*course*, *use_sis_id=False*, ***kwargs*)

Add a course to the current user's favorites. If the course is already in the user's favorites, nothing happens.

Calls POST /api/v1/users/self/favorites/courses/:id

Parameters

- **course** (*canvasapi.course.Course* or int) – The course or ID/SIS ID of the course.
- **use_sis_id** (*bool*) – Whether or not *course* is an sis ID. Defaults to *False*.

Return type *canvasapi.favorite.Favorite*

add_favorite_group(*group*, *use_sis_id=False*, ***kwargs*)

Add a group to the current user's favorites. If the group is already in the user's favorites, nothing happens.

Calls POST /api/v1/users/self/favorites/groups/:id

Parameters

- **group** (*canvasapi.group.Group* or int) – The ID or SIS ID of the group.
- **use_sis_id** (*bool*) – Whether or not *group* is an sis ID. Defaults to *False*.

Return type *canvasapi.favorite.Favorite*

create_bookmark(*name*, *url*, ***kwargs*)

Create a new Bookmark.

Calls POST /api/v1/users/self/bookmarks

Parameters

- **name** (*str*) – The name of the bookmark.
- **url** (*str*) – The url of the bookmark.

Return type *canvasapi.bookmark.Bookmark*

get_bookmark(*bookmark*)

Return single Bookmark by id

Calls GET /api/v1/users/self/bookmarks/:id

Parameters **bookmark** (*canvasapi.bookmark.Bookmark* or int) – The object or ID of the bookmark.

Return type *canvasapi.bookmark.Bookmark*

get_bookmarks(***kwargs*)

List bookmarks that the current user can view or manage.

Calls GET /api/v1/users/self/bookmarks

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.bookmark.Bookmark*

get_favorite_courses(***kwargs*)

Retrieve the paginated list of favorite courses for the current user. If the user has not chosen any favorites, then a selection of currently enrolled courses will be returned.

Calls GET /api/v1/users/self/favorites/courses

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.course.Course*

get_favorite_groups(***kwargs*)

Retrieve the paginated list of favorite groups for the current user. If the user has not chosen any favorites, then a selection of groups that the user is a member of will be returned.

Calls GET /api/v1/users/self/favorites/groups

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.Group*

get_groups(***kwargs*)

Return the list of active groups for the user.

Calls GET /api/v1/users/self/groups

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.Group*

list_bookmarks(***kwargs*)

List bookmarks that the current user can view or manage.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.current_user.CurrentUser.get_bookmarks()* instead.

Calls GET /api/v1/users/self/bookmarks

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.bookmark.Bookmark*

list_groups(***kwargs*)

Return the list of active groups for the user.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.current_user.CurrentUser.get_groups()* instead.

Calls GET /api/v1/users/self/groups

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.group.Group*

reset_favorite_courses(***kwargs*)

Reset the current user's course favorites to the default automatically generated list of enrolled courses

Calls DELETE /api/v1/users/self/favorites/courses

Returns *True* if reset correctly, *False* otherwise.

Return type bool

reset_favorite_groups(***kwargs*)

Reset the current user's group favorites to the default automatically generated list of enrolled groups

Calls DELETE /api/v1/users/self/favorites/groups

Returns *True* if reset correctly, *False* otherwise.

Return type bool

6.26 DiscussionEntry

class canvasapi.discussion_topic.**DiscussionEntry**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete this discussion entry.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:id

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:id

Return type bool

get_discussion()

Return the discussion topic object this entry is related to

Return type *canvasapi.discussion_topic.DiscussionTopic*

get_replies(***kwargs*)

Retrieves the replies to a top-level entry in a discussion topic.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/replies

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/replies

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.discussion_topic.DiscussionEntry*

list_replies(***kwargs*)

Retrieves the replies to a top-level entry in a discussion topic.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.discussion_topic.DiscussionEntry.get_replies()` instead.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/replies

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/replies

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.discussion_topic.DiscussionEntry*

mark_as_read()

Mark a discussion entry as read.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/read

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/read

Return type bool

mark_as_unread()

Mark a discussion entry as unread.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/read

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/read

Return type bool

post_reply(***kwargs*)

Add a reply to this entry.

Calls POST /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/replies

or POST /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/replies

Return type *canvasapi.discussion_topic.DiscussionEntry*

rate(*rating*, ***kwargs*)

Rate this discussion entry.

Calls POST /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:entry_id/rating

or POST /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:entry_id/rating

Parameters **rating** (*int*) – A rating to set on this entry. Only 0 and 1 are accepted.

Return type bool

update (***kwargs*)

Updates an existing discussion entry.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id/entries/:id

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id/entries/:id

Return type bool

6.27 DiscussionTopic

class canvasapi.discussion_topic.**DiscussionTopic** (*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete ()

Deletes the discussion topic. This will also delete the assignment.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id

Returns True if the discussion topic was deleted, False otherwise.

Return type bool

get_entries (*ids*, ***kwargs*)

Retrieve a paginated list of discussion entries, given a list of ids. Entries will be returned in id order, smallest id first.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entry_list

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entry_list

Parameters **ids** (*canvasapi.discussion_topic.DiscussionEntry*, or list or tuple of int) – A list of entry objects or IDs to retrieve.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.discussion_topic.DiscussionEntry*

get_parent ()

Return the object that spawned this discussion topic.

Return type *canvasapi.group.Group* or *canvasapi.course.Course*

get_topic_entries (***kwargs*)

Retreive the top-level entries in a discussion topic.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entries

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entries

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.discussion_topic.DiscussionEntry*

list_entries(*ids*, ***kwargs*)

Retrieve a paginated list of discussion entries, given a list of ids. Entries will be returned in id order, smallest id first.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi. discussion_topic. DiscussionTopic.get_entries()` instead.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entry_list

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entry_list

Parameters **ids** (*canvasapi.discussion_topic.DiscussionEntry*, or list or tuple of int) – A list of entry objects or IDs to retrieve.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.discussion_topic.DiscussionEntry*

list_topic_entries(***kwargs*)

Retreive the top-level entries in a discussion topic.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.discussion_topic.DiscussionTopic.get_topic_entries()* instead.

Calls GET /api/v1/courses/:course_id/discussion_topics/:topic_id/entries

or GET /api/v1/groups/:group_id/discussion_topics/:topic_id/entries

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.discussion_topic.DiscussionEntry*

mark_as_read()

Mark the initial text of the discussion topic as read.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id/read

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id/read

Return type bool

mark_as_unread()

Mark the initial text of the discussion topic as unread.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id/read

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id/read

Return type bool

mark_entries_as_read(***kwargs*)

Mark the discussion topic and all its entries as read.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id/read_all

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id/read_all

Return type bool

mark_entries_as_unread(***kwargs*)

Mark the discussion topic and all its entries as unread.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id/read_all

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id/read_all

Return type bool

post_entry(***kwargs*)

Creates a new entry in a discussion topic.

Calls POST /api/v1/courses/:course_id/discussion_topics/:topic_id/entries

or POST /api/v1/groups/:group_id/discussion_topics/:topic_id/entries

Return type *canvasapi.discussion_topic.DiscussionEntry*

subscribe()

Subscribe to a topic to receive notifications about new entries.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id/subscribed

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id/subscribed

Return type bool

unsubscribe()

Unsubscribe from a topic to stop receiving notifications about new entries.

Calls DELETE /api/v1/courses/:course_id/discussion_topics/:topic_id/subscribed

or DELETE /api/v1/groups/:group_id/discussion_topics/:topic_id/subscribed

Return type bool

update(***kwargs*)

Updates an existing discussion topic for the course or group.

Calls PUT /api/v1/courses/:course_id/discussion_topics/:topic_id

or PUT /api/v1/groups/:group_id/discussion_topics/:topic_id

Return type *canvasapi.discussion_topic.DiscussionTopic*

6.28 Enrollment

class canvasapi.enrollment.**Enrollment**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

deactivate(*task*)

Delete, conclude, or deactivate an enrollment.

The following tasks can be performed on an enrollment: conclude, delete, inactivate, deactivate.

Calls DELETE /api/v1/courses/:course_id/enrollments/:id

Parameters **task** (*str*) – The task to perform on the enrollment.

Return type *canvasapi.enrollment.Enrollment*

reactivate()

Activate an inactive enrollment.

Calls PUT /api/v1/courses/:course_id/enrollments/:id/reactivate

Return type *canvasapi.enrollment.Enrollment*

6.29 EnrollmentTerm

class canvasapi.enrollment_term.**EnrollmentTerm**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Delete this Enrollment Term.

Calls DELETE /api/v1/accounts/:account_id/terms/:id

Return type *canvasapi.enrollment_term.EnrollmentTerm*

edit(***kwargs*)

Modify this Enrollment Term.

Calls PUT /api/v1/accounts/:account_id/terms/:id

Return type *canvasapi.enrollment_term.EnrollmentTerm*

6.30 ExternalTool

class canvasapi.external_tool.**ExternalTool**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Remove the specified external tool.

Calls DELETE /api/v1/courses/:course_id/external_tools/:external_tool_id or DELETE /api/v1/accounts/:account_id/external_tools/:external_tool_id

Return type *canvasapi.external_tool.ExternalTool*

edit(***kwargs*)

Update the specified external tool.

Calls PUT /api/v1/courses/:course_id/external_tools/:external_tool_id or PUT /api/v1/accounts/:account_id/external_tools/:external_tool_id

Return type *canvasapi.external_tool.ExternalTool*

get_parent()

Return the object that spawned this tool.

Return type *canvasapi.account.Account* or canvasapi.account.Course

get_sessionless_launch_url(***kwargs*)

Return a sessionless launch url for an external tool.

Calls GET /api/v1/courses/:course_id/external_tools/sessionless_launch or GET /api/v1/accounts/:account_id/external_tools/sessionless_launch

Return type str

parent_id

Return the id of the course or account that spawned this tool.

Return type int

parent_type

Return whether the tool was spawned from a course or account.

Return type str

6.31 Favorite

class canvasapi.favorite.**Favorite**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

remove(***kwargs*)

Remove a course or group from the current user's favorites.

Calls

Course DELETE /api/v1/users/self/favorites/courses/:id

Group DELETE /api/v1/users/self/favorites/groups/:id

Return type *canvasapi.favorite.Favorite*

6.32 File

class canvasapi.file.**File**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Delete this file.

Calls DELETE /api/v1/files/:id

Return type *canvasapi.file.File*

download(*location*)

Download the file to specified location.

Parameters **location** (*str*) – The path to download to.

get_contents()
Download the contents of this file.

Return type str

6.33 Folder

class canvasapi.folder.**Folder**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

copy_file(*source_file*, ***kwargs*)
Copies a file into the current folder.

Calls POST /api/v1/folders/:dest_folder_id/copy_file

Parameters **source_file** (int or *canvasapi.file.File*) – The object or id of the source file.

Return type *canvasapi.folder.Folder*

create_folder(*name*, ***kwargs*)
Creates a folder within this folder.

Calls POST /api/v1/folders/:folder_id/folders

Parameters **name** (*str*) – The name of the folder.

Return type *canvasapi.folder.Folder*

delete(***kwargs*)

Remove this folder. You can only delete empty folders unless you set the 'force' flag.

Calls DELETE /api/v1/folders/:id

Return type *canvasapi.folder.Folder*

get_files(***kwargs*)
Returns the paginated list of files for the folder.

Calls GET /api/v1/folders/:id/files

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.file.File*

get_folders(***kwargs*)
Returns the paginated list of folders in the folder.

Calls GET /api/v1/folders/:id/folders

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.folder.Folder*

list_files(***kwargs*)
Returns the paginated list of files for the folder.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.folder.Folder.get_files()* instead.

Calls GET /api/v1/folders/:id/files

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.file.File*

list_folders(***kwargs*)

Returns the paginated list of folders in the folder.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.folder.Folder.get_folders()* instead.

Calls GET /api/v1/folders/:id/folders

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.folder.Folder*

update(***kwargs*)

Updates a folder.

Calls PUT /api/v1/folders/:id

Return type *canvasapi.folder.Folder*

upload(*file*, ***kwargs*)

Upload a file to this folder.

Calls POST /api/v1/folders/:folder_id/files

Parameters **file** (*file or str*) – The file or path of the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.34 GradingPeriod

class canvasapi.grading_period.**GradingPeriod**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete a grading period for a course.

Calls DELETE /api/v1/courses/:course_id/grading_periods/:id

Returns Status code 204 if delete was successful

Return type int

update (*grading_period*, ***kwargs*)
Update a grading period for a course.

Calls PUT /api/v1/courses/:course_id/grading_periods/:id

Parameters **grading_period** (*list[dict]*) – List of nested paramameters.

Return type *canvasapi.grading_period.GradingPeriod*

6.35 Group

class canvasapi.group.**Group** (*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_content_migration (*migration_type*, ***kwargs*)
Create a content migration.

Calls POST /api/v1/groups/:group_id/content_migrations

Parameters **migration_type** (str or *canvasapi.content_migration.Migrator*) – The migrator type to use in this migration

Return type *canvasapi.content_migration.ContentMigration*

create_discussion_topic (***kwargs*)
Creates a new discussion topic for the course or group.

Calls POST /api/v1/groups/:group_id/discussion_topics

Return type *canvasapi.discussion_topic.DiscussionTopic*

create_external_feed (*url*, ***kwargs*)
Create a new external feed for the group.

Calls POST /api/v1/groups/:group_id/external_feeds

Parameters **url** (*str*) – The urlof the external rss or atom feed

Return type canvasapi.external_feed.ExternalFeed

create_folder (*name*, ***kwargs*)
Creates a folder in this group.

Calls POST /api/v1/groups/:group_id/folders

Parameters **name** (*str*) – The name of the folder.

Return type *canvasapi.folder.Folder*

create_membership (*user*, ***kwargs*)
Join, or request to join, a group, depending on the join_level of the group. If the membership or join request already exists, then it is simply returned.

Calls POST /api/v1/groups/:group_id/memberships

Parameters **user** (*canvasapi.user.User* or int) – The object or ID of the user.

Return type *canvasapi.group.GroupMembership*

create_page(*wiki_page*, ***kwargs*)
Create a new wiki page.

Calls POST /api/v1/groups/:group_id/pages

Parameters **wiki_page** (*dict*) – Details about the page to create.

Returns The created page.

Return type *canvasapi.page.Page*

delete()
Delete a group.

Calls DELETE /api/v1/groups/:group_id

Return type *canvasapi.group.Group*

delete_external_feed(*feed*)
Deletes the external feed.

Calls DELETE /api/v1/groups/:group_id/external_feeds/:external_feed_id

Parameters **feed** (canvasapi.external_feed.ExternalFeed or int) – The object or id of the feed to be deleted.

Return type canvasapi.external_feed.ExternalFeed

edit(***kwargs*)
Edit a group.

Calls PUT /api/v1/groups/:group_id

Return type *canvasapi.group.Group*

edit_front_page(***kwargs*)
Update the title or contents of the front page.

Calls PUT /api/v1/groups/:group_id/front_page

Return type *canvasapi.page.Page*

export_content(*export_type*, ***kwargs*)
Begin a content export job for a group.

Calls POST /api/v1/groups/:group_id/content_exports

Parameters **export_type** (*str*) – The type of content to export.

Return type *canvasapi.content_export.ContentExport*

get_activity_stream_summary()
Return a summary of the current user's global activity stream.

Calls GET /api/v1/groups/:group_id/activity_stream/summary

Return type dict

get_assignment_override(*assignment*, ***kwargs*)
Return override for the specified assignment for this group.

Parameters **assignment** (*canvasapi.assignment.Assignment* or int) – The assignment to get an override for

Calls GET /api/v1/groups/:group_id/assignments/:assignment_id/override

Return type canvasapi.assignment.AssignmentOverride

get_content_export (*content_export*, ***kwargs*)
Return information about a single content export.

Calls GET /api/v1/groups/:group_id/content_exports/:id

Parameters **content_export** (int or *canvasapi.content_export.ContentExport*) – The object or ID of the content export to show.

Return type *canvasapi.content_export.ContentExport*

get_content_exports (***kwargs*)
Return a paginated list of the past and pending content export jobs for a group.

Calls GET /api/v1/groups/:group_id/content_exports

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_export.ContentExport*

get_content_migration (*content_migration*, ***kwargs*)
Retrive a content migration by its ID

Calls GET /api/v1/groups/:group_id/content_migrations/:id

Parameters **content_migration** (int, str or *canvasapi.content_migration.ContentMigration*) – The object or ID of the content migration to retrieve.

Return type *canvasapi.content_migration.ContentMigration*

get_content_migrations (***kwargs*)
List content migrations that the current account can view or manage.

Calls GET /api/v1/groups/:group_id/content_migrations/

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_migration.ContentMigration*

get_discussion_topic (*topic*)
Return data on an individual discussion topic.

Calls GET /api/v1/groups/:group_id/discussion_topics/:topic_id

Parameters **topic** (*canvasapi.discussion_topic.DiscussionTopic* or int) – The object or ID of the discussion topic.

Return type *canvasapi.discussion_topic.DiscussionTopic*

get_discussion_topics (***kwargs*)
Returns the paginated list of discussion topics for this course or group.

Calls GET /api/v1/groups/:group_id/discussion_topics

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.discussion_topic.DiscussionTopic*

get_external_feeds (***kwargs*)
Returns the list of External Feeds this group.

Calls GET /api/v1/groups/:group_id/external_feeds

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.external_feed.ExternalFeed`

get_file (*file*, ***kwargs*)
Return the standard attachment json object for a file.

Calls GET /api/v1/groups/:group_id/files/:id

Parameters **file** (*canvasapi.file.File* or int) – The object or ID of the file to retrieve.

Return type *canvasapi.file.File*

get_files(***kwargs*)

Returns the paginated list of files for the group.

Calls GET /api/v1/groups/:group_id/files

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.file.File*

get_folder(*folder*)

Returns the details for a group's folder

Calls GET /api/v1/groups/:group_id/folders/:id

Parameters **folder** (*canvasapi.folder.Folder* or int) – The object or ID of the folder to retrieve.

Return type *canvasapi.folder.Folder*

get_folders(***kwargs*)

Returns the paginated list of all folders for the given group. This will be returned as a flat list containing all subfolders as well.

Calls GET /api/v1/groups/:group_id/folders

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.folder.Folder*

get_full_discussion_topic(*topic*)

Return a cached structure of the discussion topic.

Calls GET /api/v1/groups/:group_id/discussion_topics/:topic_id/view

Parameters **topic** (*canvasapi.discussion_topic.DiscussionTopic* or int) – The object or ID of the discussion topic.

Return type dict

get_membership(*user*, *membership_type*)

List users in a group.

Calls GET /api/v1/groups/:group_id/users/:user_id

or GET /api/v1/groups/:group_id/memberships/:membership_id

Parameters **user** (*canvasapi.user.User* or int) – list of user ids

Return type *canvasapi.group.GroupMembership*

get_memberships(***kwargs*)

List users in a group.

Calls GET /api/v1/groups/:group_id/memberships

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.group.GroupMembership*

get_migration_systems(***kwargs*)

Return a list of migration systems.

Calls GET /api/v1/groups/:group_id/content_migrations/migrators

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.content_migration.Migrator*

get_page(*url*)

Retrieve the contents of a wiki page.

Calls GET /api/v1/groups/:group_id/pages/:url

Parameters **url** (*str*) – The url for the page.

Returns The specified page.

Return type `canvasapi.groups.Group`

get_pages(***kwargs*)

List the wiki pages associated with a group.

Calls GET /api/v1/groups/:group_id/pages

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.page.Page*

get_tabs(***kwargs*)

List available tabs for a group. Returns a list of navigation tabs available in the current context.

Calls GET /api/v1/groups/:group_id/tabs

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.tab.Tab*

get_users(***kwargs*)

List users in a group.

Calls GET /api/v1/groups/:group_id/users

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.user.User*

invite(*invitees*)

Invite users to group.

Calls POST /api/v1/groups/:group_id/invite

Parameters **invitees** (*integer list*) – list of user ids

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.group.GroupMembership*

list_external_feeds(***kwargs*)

Returns the list of External Feeds this group.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.group.Group.get_external_feeds()* instead.

Calls GET /api/v1/groups/:group_id/external_feeds

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.external_feed.ExternalFeed`

list_files(***kwargs*)

Returns the paginated list of files for the group.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.group.Group.get_files()* instead.

Calls GET /api/v1/groups/:group_id/files

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.file.File`*

`list_folders`(***kwargs*)

Returns the paginated list of all folders for the given group. This will be returned as a flat list containing all subfolders as well.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.Group.get_folders()`* instead.

Calls GET /api/v1/groups/:group_id/folders

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.folder.Folder`*

`list_memberships`(***kwargs*)

List users in a group.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.Group.get_memberships()`* instead.

Calls GET /api/v1/groups/:group_id/memberships

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.group.GroupMembership`*

`list_tabs`(***kwargs*)

List available tabs for a group. Returns a list of navigation tabs available in the current context.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.Group.get_tabs()`* instead.

Calls GET /api/v1/groups/:group_id/tabs

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.tab.Tab`*

`list_users`(***kwargs*)

List users in a group.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.Group.get_users()`* instead.

Calls GET /api/v1/groups/:group_id/users

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.user.User`*

preview_html(*html*)

Preview HTML content processed for this course.

Calls POST /api/v1/groups/:group_id/preview_html

Parameters **html** (*str*) – The HTML code to preview.

Return type str

remove_user(*user*)

Leave a group if allowed.

Calls DELETE /api/v1/groups/:group_id/users/:user_id

Parameters **user** (*canvasapi.user.User* or int) – The user object or ID to remove from the group.

Return type *canvasapi.user.User*

reorder_pinned_topics(*order*)

Puts the pinned discussion topics in the specified order. All pinned topics should be included.

Calls POST /api/v1/groups/:group_id/discussion_topics/reorder

Parameters **order** (*iterable sequence of values*) – The ids of the pinned discussion topics in the desired order. e.g. [104, 102, 103]

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.discussion_topic.DiscussionTopic*

show_front_page()

Retrieve the content of the front page.

Calls GET /api/v1/groups/:group_id/front_page

Return type *canvasapi.group.Group*

update_membership(*user*, ***kwargs*)

Accept a membership request, or add/remove moderator rights.

Calls PUT /api/v1/groups/:group_id/users/:user_id

Parameters **user** (*canvasapi.user.User* or int) – The object or ID of the user.

Return type *canvasapi.group.GroupMembership*

upload(*file*, ***kwargs*)

Upload a file to the group. Only those with the 'Manage Files' permission on a group can upload files to the group. By default, this is anybody participating in the group, or any admin over the group.

Calls POST /api/v1/groups/:group_id/files

Parameters

- **path** (*str*) – The path of the file to upload.
- **file** (*file or str*) – The file or path of the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.36 GroupMembership

class canvasapi.group.**GroupMembership**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

remove_self()

Leave a group if allowed.

Calls DELETE /api/v1/groups/:group_id/memberships/:membership_id

Returns An empty dictionary

Return type dict

remove_user(*user*)

Remove user from membership.

Calls DELETE /api/v1/groups/:group_id/users/:user_id

Parameters user (*canvasapi.user.User* or int) – The user object or ID to remove from the group.

Returns An empty dictionary

Return type dict

update(***kwargs*)

Accept a membership request, or add/remove moderator rights.

Calls PUT /api/v1/groups/:group_id/memberships/:membership_id

Return type *canvasapi.group.GroupMembership*

6.37 GroupCategory

class canvasapi.group.**GroupCategory**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

assign_members(*sync=False*)

Assign unassigned members.

Calls POST /api/v1/group_categories/:group_category_id/assign_unassigned_members

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.user.User* or *canvasapi.progress.Progress*

create_group(***kwargs*)

Create a group.

Calls POST /api/v1/group_categories/:group_category_id/groups

Return type *canvasapi.group.Group*

delete()

Delete a group category.

Calls DELETE /api/v1/group_categories/:group_category_id

Return type empty dict

get_groups(***kwargs*)

List groups in group category.

Calls GET /api/v1/group_categories/:group_category_id/groups

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.group.Group`*

get_users(***kwargs*)

List users in group category.

Calls GET /api/v1/group_categories/:group_category_id/users

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.user.User`*

list_groups(***kwargs*)

List groups in group category.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.GroupCategory.get_groups()`* instead.

Calls GET /api/v1/group_categories/:group_category_id/groups

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.group.Group`*

list_users(***kwargs*)

List users in group category.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.group.GroupCategory.get_users()`* instead.

Calls GET /api/v1/group_categories/:group_category_id/users

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.user.User`*

update(***kwargs*)

Update a group category.

Calls PUT /api/v1/group_categories/:group_category_id

Return type *`canvasapi.group.GroupCategory`*

6.38 Login

class canvasapi.login.**Login**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete()

Delete an existing login.

Calls DELETE /api/v1/users/:user_id/logins/:id

Return type *canvasapi.login.Login*

edit(***kwargs*)

Update an existing login for a user in the given account.

Calls PUT /api/v1/accounts/:account_id/logins/:id

Return type *canvasapi.login.Login*

6.39 Module

class canvasapi.module.**Module**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_module_item(*module_item*, ***kwargs*)

Create a module item.

Calls POST /api/v1/courses/:course_id/modules/:module_id/items

Parameters **module_item** (*dict*) – The attributes to create the module item with.

Returns The created module item.

Return type *canvasapi.module.ModuleItem*

delete()

Delete this module.

Calls DELETE /api/v1/courses/:course_id/modules/:id

Return type *canvasapi.module.Module*

edit(***kwargs*)

Update this module.

Calls PUT /api/v1/courses/:course_id/modules/:id

Return type *canvasapi.module.Module*

get_module_item(*module_item*, ***kwargs*)

Retrieve a module item by ID.

Calls GET /api/v1/courses/:course_id/modules/:module_id/items/:id

Parameters **module_item** (*canvasapi.module.ModuleItem* or dict) – The object or ID of the module item.

Return type *canvasapi.module.ModuleItem*

get_module_items(***kwargs*)

List all of the items in this module.

Calls GET /api/v1/courses/:course_id/modules/:module_id/items

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.module.ModuleItem*

list_module_items(***kwargs*)

List all of the items in this module.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.module.Modules.get_module_items()` instead.

Calls GET /api/v1/courses/:course_id/modules/:module_id/items

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.module.ModuleItem*

relock()

Reset module progressions to their default locked state and recalculates them based on the current requirements.

Adding progression requirements to an active course will not lock students out of modules they have already unlocked unless this action is called.

Calls PUT /api/v1/courses/:course_id/modules/:id/relock

Return type *canvasapi.module.Module*

6.40 ModuleItem

class `canvasapi.module.`**ModuleItem**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

complete()

Mark this module item as done.

Calls PUT /api/v1/courses/:course_id/modules/:module_id/items/:id/done

Return type *canvasapi.module.ModuleItem*

delete()

Delete this module item.

Calls DELETE /api/v1/courses/:course_id/modules/:module_id/items/:id

Return type *canvasapi.module.ModuleItem*

edit(***kwargs*)

Update this module item.

Calls PUT /api/v1/courses/:course_id/modules/:module_id/items/:id

Returns The updated module item.

Return type *canvasapi.module.ModuleItem*

uncomplete()

Mark this module item as not done.

Calls DELETE /api/v1/courses/:course_id/modules/:module_id/items/:id/done

Return type *canvasapi.module.ModuleItem*

6.41 Outcome

class canvasapi.outcome.**Outcome**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

update(***kwargs*)

Modify an existing outcome.

Calls PUT /api/v1/outcomes/:id

Returns True if updated, False otherwise.

Return type bool

6.42 OutcomeLink

class canvasapi.outcome.**OutcomeLink**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

get_outcome()

Return the linked outcome

Calls GET /api/v1/outcomes/:id

Returns Outcome object that was in the OutcomeLink

Return type *canvasapi.outcome.Outcome*

get_outcome_group()

Return the linked outcome group

Calls GET /api/v1/global/outcome_groups/:id or GET /api/v1/accounts/:account_id/outcome_groups/:id or GET /api/v1/courses/:course_id/outcome_groups/:id

Returns Linked outcome group object.

Return type *canvasapi.outcome.OutcomeGroup*

6.43 OutcomeGroup

class canvasapi.outcome.**OutcomeGroup**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_subgroup(*title*, ***kwargs*)

Create a subgroup of the current group

Calls POST /api/v1/global/outcome_groups/:id/subgroups or POST /api/v1/accounts/:account_id/outcome_groups/:id/subgroups or POST /api/v1/courses/:course_id/outcome_groups/:id/subgroups

Parameters **title** (*str*) – The title of the subgroup.

Returns Itself as an OutcomeGroup object.

Return type *canvasapi.outcome.OutcomeGroup*

delete()

Delete an outcome group.

Calls DELETE /api/v1/global/outcome_groups/:id or DELETE /api/v1/accounts/:account_id/outcome_groups/:id or DELETE /api/v1/courses/:course_id/outcome_groups/:id

Returns True if successful, false if failed.

Return type bool

get_linked_outcomes(***kwargs*)

List linked outcomes.

Calls GET /api/v1/global/outcome_groups/:id/outcomes or GET /api/v1/accounts/:account_id/outcome_groups/:id/outcomes or GET /api/v1/courses/:course_id/outcome_groups/:id/outcomes

Returns Paginated List of Outcomes linked to the group.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.outcome.OutcomeLink*

get_subgroups(***kwargs*)

List subgroups.

Calls GET /api/v1/global/outcome_groups/:id/subgroups or GET /api/v1/accounts/:account_id/outcome_groups/:id/subgroups or GET /api/v1/courses/:course_id/outcome_groups/:id/subgroups

Returns Paginated List of OutcomeGroups linked to the current group.

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.outcome.OutcomeGroup*

import_outcome_group(*outcome_group*)

Import an outcome group as a subgroup into the current outcome group

Calls POST /api/v1/global/outcome_groups/:id/import or POST /api/v1/accounts/:account_id/outcome_groups/:id/import or POST /api/v1/courses/:course_id/outcome_groups/:id/import

Parameters **outcome** (*canvasapi.outcome.OutcomeGroup* or int) – The object or ID of the outcome group to import.

Returns Itself as an OutcomeGroup object.

Return type *canvasapi.outcome.OutcomeGroup*

link_existing(*outcome*)

Link to an existing Outcome.

Calls PUT /api/v1/global/outcome_groups/:id/outcomes/:outcome_id or PUT /api/v1/accounts/:account_id/outcome_groups/:id/outcomes/:outcome_id or PUT /api/v1/courses/:course_id/outcome_groups/:id/outcomes/:outcome_id

Parameters **outcome** (*canvasapi.outcome.Outcome* or int) – The object or ID of the outcome.

Returns OutcomeLink object with current OutcomeGroup and newly linked Outcome.

Return type *canvasapi.outcome.OutcomeLink*

link_new(*title*, ***kwargs*)

Create a new Outcome and link it to this OutcomeGroup

Calls POST /api/v1/global/outcome_groups/:id/outcomes or POST /api/v1/accounts/:account_id/outcome_groups/:id/outcomes or POST /api/v1/courses/:course_id/outcome_groups/:id/outcomes

Parameters **title** (*str*) – The title of the new outcome.

Returns OutcomeLink object with current OutcomeGroup and newly linked Outcome.

Return type *canvasapi.outcome.OutcomeLink*

list_linked_outcomes(***kwargs*)

List linked outcomes.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.outcome.Outcome.get_linked_outcomes()` instead.

Calls GET /api/v1/global/outcome_groups/:id/outcomes or GET /api/v1/accounts/:account_id/outcome_groups/:id/outcomes or GET /api/v1/courses/:course_id/outcome_groups/:id/outcomes

Returns Paginated List of Outcomes linked to the group.

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.outcome.OutcomeLink*

list_subgroups(***kwargs*)

List subgroups.

> **Warning:** Deprecated since version 0.10.0: Use `canvasapi.outcome.Outcome.get_subgroups()` instead.

Calls GET /api/v1/global/outcome_groups/:id/subgroups or GET /api/v1/accounts/:account_id/outcome_groups/:id/subgroups or GET /api/v1/courses/:course_id/outcome_groups/:id/subgroups

Returns Paginated List of OutcomeGroups linked to the current group.

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.outcome.OutcomeGroup`*

`unlink_outcome`(*outcome*)

Remove an Outcome from and OutcomeLink

Calls DELETE /api/v1/global/outcome_groups/:id/outcomes/:outcome_id or DELETE /api/v1/accounts/:account_id/outcome_groups/:id/outcomes/:outcome_id or DELETE /api/v1/courses/:course_id/outcome_groups/:id/outcomes/:outcome_id

Parameters **`outcome`** (*`canvasapi.outcome.Outcome`* or int) – The object or ID of the outcome.

Returns True if successful, false if failed.

Return type bool

`update`(***kwargs*)

Update an outcome group.

Calls PUT /api/v1/global/outcome_groups/:id or PUT /api/v1/accounts/:account_id/outcome_groups/:id or PUT /api/v1/courses/:course_id/outcome_groups/:id

Returns True if updated, False otherwise.

Return type bool

6.44 OutcomeImport

class `canvasapi.outcome_import.`**`OutcomeImport`**(*requester*, *attributes*)

Parameters

- **`requester`** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **`attributes`** (*`dict`*) – The JSON object to build this object with.

6.45 Page

class `canvasapi.page.`**`Page`**(*requester*, *attributes*)

Parameters

- **`requester`** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **`attributes`** (*`dict`*) – The JSON object to build this object with.

`delete`()

Delete this page.

Calls DELETE /api/v1/courses/:course_id/pages/:url

Return type *canvasapi.page.Page*

edit(***kwargs*)
Update the title or the contents of a specified wiki page.

Calls PUT /api/v1/courses/:course_id/pages/:url

Return type *canvasapi.page.Page*

get_parent()
Return the object that spawned this page.

Calls GET /api/v1/groups/:group_id or :calls: GET /api/v1/courses/:id

Return type *canvasapi.group.Group* or *canvasapi.course.Course*

get_revision_by_id(*revision*, ***kwargs*)
Retrieve the contents of the revision by the id.

Calls GET /api/v1/courses/:course_id/pages/:url/revisions/:revision_id

Parameters **revision** (`canvasapi.pagerevision.PageRevision` or int) – The object or ID of a specified revision.

Returns Contents of the page revision.

Return type `canvasapi.pagerevision.PageRevision`

get_revisions(***kwargs*)
List the revisions of a page.

Calls GET /api/v1/courses/:course_id/pages/:url/revisions

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.pagerevision.PageRevision`

list_revisions(***kwargs*)
List the revisions of a page.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.page.Page.get_revisions()* instead.

Calls GET /api/v1/courses/:course_id/pages/:url/revisions

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.pagerevision.PageRevision`

parent_id
Return the id of the course or group that spawned this page.

Return type int

parent_type
Return whether the page was spawned from a course or group.

Return type str

revert_to_revision(*revision*)
Revert the page back to a specified revision.

Calls POST /api/v1/courses/:course_id/pages/:url/revisions/:revision_id

Parameters **revision** (`canvasapi.pagerevision.PageRevision` or int) – The object or ID of a specified revision.

Returns Contents of the page revision.

Return type `canvasapi.pagerevision.PageRevision`

show_latest_revision(***kwargs*)
Retrieve the contents of the latest revision.

Calls GET /api/v1/courses/:course_id/pages/:url/revisions/latest

Return type `canvasapi.pagerevision.PageRevision`

6.46 PageRevision

class canvasapi.page.**PageRevision**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

get_parent()
Return the object that spawned this page.

Calls GET /api/v1/groups/:group_id or :calls: GET /api/v1/courses/:id

Return type *canvasapi.group.Group* or *canvasapi.course.Course*

parent_id
Return the id of the course or group that spawned this page.

Return type int

parent_type
Return whether the page was spawned from a course or group.

Return type str

6.47 PlannerNote

class canvasapi.planner.**PlannerNote**(*requester*, *attributes*)

Parameters

- **requester** (`canvasapi.requester.Requester`) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)
Delete a planner note for the current user

Calls DELETE /api/v1/planner_notes/:id

Return type *canvasapi.planner.PlannerNote*

update(***kwargs*)
Update a planner note for the current user

Calls PUT /api/v1/planner_notes/:id

Return type *canvasapi.planner.PlannerNote*

6.48 PlannerOverride

class canvasapi.planner.**PlannerOverride**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete a planner override for the current user

Calls DELETE /api/v1/planner/overrides/:id

Return type *canvasapi.planner.PlannerOverride*

update(***kwargs*)

Update a planner override's visibilty for the current user

Calls PUT /api/v1/planner/overrides/:id

Return type *canvasapi.planner.PlannerOverride*

6.49 Poll

class canvasapi.poll.**Poll**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_choice(*poll_choice*, ***kwargs*)

Create a new choice for the current poll.

Calls POST /api/v1/polls/:poll_id/poll_choices

Parameters poll_choice (*list*) – 'text' is required, 'is_correct' and 'position' are optional.

Return type *canvasapi.poll_choice.PollChoice*

create_session(*poll_session*, ***kwargs*)

Create a new poll session for this poll

Calls POST /api/v1/polls/:poll_id/poll_sessions

Parameters poll_session (*list*) – List of arguments. course_id (required): id of the course for the session, course_section_id (optional): id of the course section for this session, has_public_results (optional): whether the results are viewable by students.

Return type *canvasapi.poll_session.PollSession*

delete(***kwargs*)

Delete a single poll, based on the poll id.

Calls DELETE /api/v1/polls/:id

Returns True if the deletion was successful, false otherwise.

Return type bool

get_choice(*poll_choice*, ***kwargs*)
Returns the poll choice with the given id.

Calls GET /api/v1/polls/:poll_id/poll_choices/:id

Return type *canvasapi.poll_choice.PollChoice*

get_choices(***kwargs*)
Returns a paginated list of PollChoices of a poll, based on poll id.

Calls GET /api/v1/polls/:poll_id/poll_choices

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.poll_choice.PollChoice*

get_session(*poll_session*, ***kwargs*)
Returns the poll session with the given id.

Calls GET /api/v1/polls/:poll_id/poll_sessions/:id

Parameters **poll_session** – List of arguments. Takes a poll session id (int) or poll session object.

Return type *canvasapi.poll_session.PollSession*

get_sessions(***kwargs*)
Returns the paginated list of PollSessions in a poll.

Calls GET /api/v1/polls/:poll_id/poll_sessions

Return type canvasapi.paginated_lsit.Paginated List of *canvasapi.poll_session.PollSession*

update(*poll*, ***kwargs*)
Update an existing poll belonging to the current user.

Calls PUT /api/v1/polls/:id

Parameters **poll** (*list*) – List of arguments. 'Question' is required and 'Description' is optional

Return type *canvasapi.poll.Poll*

6.50 PollChoice

class canvasapi.poll_choice.**PollChoice**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)
Delete a single poll, based on the poll id.

Calls DELETE /api/v1/polls/:poll_id/poll_choices/:id

Returns True if the deletion was successful, false otherwise.

Return type bool

update (*poll_choice*, ***kwargs*)

Update an existing choice for this poll.

Calls PUT /api/v1/polls/:poll_id/poll_choices/:id

Parameters **poll_choice** (*list*) – List of arguments. 'text' is required and 'is_correct' and 'position' are optional.

Return type *canvasapi.poll_choice.PollChoice*

6.51 PollSession

class canvasapi.poll_session.**PollSession** (*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

close (***kwargs*)

Close a poll session to answers based on the poll id.

Calls GET /api/v1/polls/:poll_id/poll_sessions/:id/close

Returns *canvasapi.poll_session.PollSession*

create_submission (*poll_submissions*, ***kwargs*)

Create a new poll submission for this poll session.

Calls POST /api/v1/polls/:poll_id/poll_sessions/:poll_session_id/poll_submissions

Parameters **poll_submissions** (*list*) – List of arguments. poll_choice_id (required int): Chosen poll choice for this submission.

Return type *canvasapi.poll_submission.PollSubmission*

delete (***kwargs*)

Delete a single poll session, based on the session id.

Calls DELETE /api/v1/polls/:poll_id/poll_sessions/:id

Returns True if the deletion was successful, false otherwise.

Return type bool

get_submission (*poll_submission*, ***kwargs*)

Returns the poll submission with the given id.

Calls GET /api/v1/polls/:poll_id/poll_sessions/:poll_session_id/poll_submissions/:id

Parameters **poll_submission** (int or *canvasapi.poll_submission.PollSubmission*) – Takes a poll submission id (int) or object.

Return type *canvasapi.poll_submission.PollSubmission*

open (***kwargs*)

Open a poll session to answers based on the poll id.

Calls GET /api/v1/polls/:poll_id/poll_sessions/:id/open

Returns *canvasapi.poll_session.PollSession*

update(*poll_session*, ***kwargs*)

Update an existing poll session for a poll based on poll id.

Calls PUT /api/v1/polls/:poll_id/poll_sessions/:id

Parameters **poll_session** (*list*) – List of arguments. course_id (required): id of the course for the session, course_section_id (optional): id of the course section for this session, has_public_results (optional): whether the results are viewable by students.

Return type *canvasapi.poll_session.PollSession*

6.52 PollSubmission

class canvasapi.poll_submission.**PollSubmission**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.53 Progress

class canvasapi.progress.**Progress**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

query()

Return completion and status information about an asynchronous job.

Calls GET /api/v1/progress/:id

Return type *canvasapi.progress.Progress*

6.54 Quiz

class canvasapi.quiz.**Quiz**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_question(***kwargs*)

Create a new quiz question for this quiz.

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/questions

Return type *canvasapi.quiz.QuizQuestion*

create_question_group(*quiz_groups*, ***kwargs*)
Create a new question group for the given quiz id

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/groups

Parameters **quiz_groups** (*list[dict]*) – The name, pick count, question points, and/or assessment question bank id. All of these parameters are optional, but at least one must exist (even if empty) to receive a response. The request expects a list, but will only create 1 question group per request.

Returns *QuizGroup* object

Return type canvasapi.quiz_group.QuizGroup

create_submission(***kwargs*)
Start taking a Quiz by creating a QuizSubmission can be used to answer questions and submit answers.

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/submissions

Return type *canvasapi.quiz.QuizSubmission*

delete(***kwargs*)
Delete this quiz.

Calls DELETE /api/v1/courses/:course_id/quizzes/:id

Return type *canvasapi.quiz.Quiz*

edit(***kwargs*)
Modify this quiz.

Calls PUT /api/v1/courses/:course_id/quizzes/:id

Returns The updated quiz.

Return type *canvasapi.quiz.Quiz*

get_all_quiz_submissions(***kwargs*)
Get a list of all submissions for this quiz.

> **Warning:** Deprecated since version 0.13.0: Use *canvasapi.quiz.Quiz.get_submissions()* instead.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/submissions

Return type list of *canvasapi.quiz.QuizSubmission*

get_question(*question*, ***kwargs*)
Get as single quiz question by ID.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/questions/:id

Parameters **question** (int, str or *canvasapi.quiz.QuizQuestion*) – The object or ID of the quiz question to retrieve.

Return type *canvasapi.quiz.QuizQuestion*

get_questions(***kwargs*)
List all questions for a quiz.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/questions

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.quiz.QuizQuestion`*

get_quiz_group(*id*, ***kwargs*)
Get details of the quiz group with the given id

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/groups/:id

Parameters **id** (*int*) – The ID of the question group.

Returns *QuizGroup* object

Return type `canvasapi.quiz_group.QuizGroup`

get_quiz_submission(*quiz_submission*, ***kwargs*)
Get a single quiz submission.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/submissions/:id

Parameters **quiz_submission** (int, string, *`canvasapi.quiz.QuizSubmission`*) – The object or ID of the quiz submission to retrieve.

Return type *`canvasapi.quiz.QuizSubmission`*

get_submissions(***kwargs*)
Get a list of all submissions for this quiz.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/submissions

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.quiz.QuizSubmission`*

set_extensions(*quiz_extensions*, ***kwargs*)
Set extensions for student quiz submissions.

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/extensions

Parameters **quiz_extensions** (*list*) – List of dictionaries representing extensions.

Return type list of *`canvasapi.quiz.QuizExtension`*

Example Usage:

```
>>> quiz.set_extensions([
...     {
...         'user_id': 1,
...         'extra_time': 60,
...         'extra_attempts': 1
...     },
...     {
...         'user_id': 2,
...         'extra_attempts': 3
...     },
...     {
...         'user_id': 3,
...         'extra_time': 20
...     }
... ])
```

6.55 QuizExtension

class canvasapi.quiz.**QuizExtension**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.56 QuizQuestion

class canvasapi.quiz.**QuizQuestion**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete(***kwargs*)

Delete an existing quiz question.

Calls DELETE /api/v1/courses/:course_id/quizzes/:quiz_id/questions/:id

Returns True if question was successfully deleted; False otherwise.

Return type bool

edit(***kwargs*)

Update an existing quiz question.

Calls PUT /api/v1/courses/:course_id/quizzes/:quiz_id/questions/:id

Return type *canvasapi.quiz.QuizQuestion*

6.57 QuizSubmission

class canvasapi.quiz.**QuizSubmission**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

answer_submission_questions(*validation_token=None*, ***kwargs*)

Provide or update an answer to one or more quiz questions.

Calls POST /api/v1/quiz_submissions/:quiz_submission_id/questions

Parameters **validation_token** (*str*) – (Optional) The unique validation token for this quiz submission. If one is not provided, canvasapi will attempt to use *self.validation_token*.

Returns A list of quiz submission questions.

Return type list of canvasapi.quiz.QuizSubmissionQuestion

complete(*validation_token=None*, ***kwargs*)

Complete the quiz submission by marking it as complete and grading it. When the quiz submission has been marked as complete, no further modifications will be allowed.

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/submissions/:id/complete

Parameters **validation_token** (*str*) – (Optional) The unique validation token for this quiz submission. If one is not provided, canvasapi will attempt to use *self.validation_token*.

Return type *canvasapi.quiz.QuizSubmission*

get_submission_questions (***kwargs*)

Get a list of all the question records for this quiz submission.

Calls GET /api/v1/quiz_submissions/:quiz_submission_id/questions

Returns A list of quiz submission questions.

Return type list of canvasapi.quiz.QuizSubmissionQuestion

get_times (***kwargs*)

Get the current timing data for the quiz attempt, both the end_at timestamp and the time_left parameter.

Calls GET /api/v1/courses/:course_id/quizzes/:quiz_id/submissions/:id/time

Return type dict

update_score_and_comments (***kwargs*)

Update the amount of points a student has scored for questions they've answered, provide comments for the student about their answer(s), or simply fudge the total score by a specific amount of points.

Calls PUT /api/v1/courses/:course_id/quizzes/:quiz_id/submissions/:id

Returns The updated quiz.

Return type *canvasapi.quiz.QuizSubmission*

6.58 QuizGroup

class canvasapi.quiz.**QuizGroup** (*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

delete (*id*)

Get details of the quiz group with the given id.

Calls DELETE /api/v1/courses/:course_id/quizzes/:quiz_id/groups/:id

Parameters **id** (*int*) – The ID of the question group.

Returns True if the result was successful (Status code of 204)

Return type bool

reorder_question_group (*id*, *order*, ***kwargs*)

Update the order of questions within a given group

Calls POST /api/v1/courses/:course_id/quizzes/:quiz_id/groups/:id/reorder

Parameters

- **id** (*list[dict]*) – The ID of the question group.
- **order** – A list of dictionaries containing the key 'id' of the question to be placed at order's index.

Returns True if the result was successful (Status code of 204)

Return type bool

update(*id*, *quiz_groups*, ***kwargs*)

Update a question group given by id.

Calls PUT /api/v1/courses/:course_id/quizzes/:quiz_id/groups/:id

Parameters

- **id** (*int*) – The ID of the question group.
- **quiz_groups** (*list[dict]*) – The name, pick count, and/or question points. All of these parameters are optional, but at least one must exist (even if empty) to recieve a response. The request expects a list, but will only update 1 question group per request.

Returns *True* if the QuizGroup was updated. *False* otherwise.

Return type bool

6.59 Rubric

class canvasapi.rubric.**Rubric**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

6.60 Section

class canvasapi.section.**Section**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

cross_list_section(*new_course*)

Move the Section to another course.

Calls POST /api/v1/sections/:id/crosslist/:new_course_id

Parameters **new_course** (*canvasapi.course.Course* or int) – The object or ID of the new course.

Return type *canvasapi.section.Section*

decross_list_section()

Undo cross-listing of a section.

Calls DELETE /api/v1/sections/:id/crosslist

Return type *canvasapi.section.Section*

delete()

Delete a target section.

Calls DELETE /api/v1/sections/:id

Return type *canvasapi.section.Section*

edit(***kwargs*)

Edit contents of a target section.

Calls PUT /api/v1/sections/:id

Return type *canvasapi.section.Section*

get_assignment_override(*assignment*, ***kwargs*)

Return override for the specified assignment for this section.

Parameters **assignment** (*canvasapi.assignment.Assignment* or int) – The assignment to get an override for

Calls GET /api/v1/sections/:course_section_id/assignments/:assignment_id/override

Return type canvasapi.assignment.AssignmentOverride

get_enrollments(***kwargs*)

List all of the enrollments for the current user.

Calls GET /api/v1/sections/:section_id/enrollments

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.enrollment.Enrollment*

get_multiple_submissions(***kwargs*)

List submissions for multiple assignments. Get all existing submissions for a given set of students and assignments.

Calls GET /api/v1/sections/:section_id/students/submissions

Return type canvasapi.paginated_list.PaginatedList of *canvasapi.submission.Submission*

get_submission(*assignment*, *user*, ***kwargs*)

Get a single submission, based on user id.

> **Warning:** Deprecated since version 0.9.0: Use *canvasapi.assignment.Assignment.get_submission()* instead.

Calls GET /api/v1/sections/:section_id/assignments/:assignment_id/submissions/:user_id

Parameters

- **assignment** (*canvasapi.assignment.Assignment* or int) – The object or ID of the assignment.
- **user** (*canvasapi.user.User* or int or str) – The object or ID of the user.

Return type *canvasapi.submission.Submission*

list_multiple_submissions(***kwargs*)

List submissions for multiple assignments. Get all existing submissions for a given set of students and assignments.

> **Warning:** Deprecated since version 0.10.0: Use *canvasapi.section.Section.get_multiple_submissions()* instead.

Calls GET /api/v1/sections/:section_id/students/submissions

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.submission.Submission`*

`list_submissions`(*assignment*, ***kwargs*)
Get all existing submissions for an assignment.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.assignment.Assignment.get_submissions()`* instead.

Calls GET /api/v1/sections/:section_id/assignments/:assignment_id/submissions

Parameters **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the assignment.

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.submission.Submission`*

`mark_submission_as_read`(*assignment*, *user*, ***kwargs*)
Mark submission as read. No request fields are necessary.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.submission.Submission.mark_read()`* instead.

Calls PUT /api/v1/sections/:section_id/assignments/:assignment_id/submissions/:user_id/read

Parameters

- **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the assignment.
- **`user`** (*`canvasapi.user.User`* or int or str) – The object or ID of the user.

Return type *bool*

`mark_submission_as_unread`(*assignment*, *user*, ***kwargs*)
Mark submission as unread. No request fields are necessary.

> **Warning:** Deprecated since version 0.9.0: Use *`canvasapi.submission.Submission.mark_unread()`* instead.

Calls DELETE /api/v1/sections/:section_id/assignments/:assignment_id/submissions/:user_id/read

Parameters

- **`assignment`** (*`canvasapi.assignment.Assignment`* or int) – The object or ID of the assignment.
- **`user`** (*`canvasapi.user.User`* or int or str) – The object or ID of the user.

Return type *bool*

`submissions_bulk_update`(***kwargs*)
Update the grading and comments on multiple student’s assignment submissions in an asynchronous job.

Calls POST /api/v1/sections/:section_id/submissions/update_grades

Return type *canvasapi.progress.Progress*

submit_assignment(*assignment*, *submission*, ***kwargs*)

Makes a submission for an assignment.

> **Warning:** Deprecated since version 0.9.0: Use *canvasapi.assignment.Assignment.submit()* instead.

Calls POST /api/v1/sections/:section_id/assignments/:assignment_id/submissions

Parameters

- **assignment** (*canvasapi.assignment.Assignment* or int) – The object or ID of the assignment.
- **submission** (*dict*) – The attributes of the submission.

Return type *canvasapi.submission.Submission*

update_submission(*assignment*, *user*, ***kwargs*)

Comment on and/or update the grading for a student's assignment submission.

> **Warning:** Deprecated since version 0.9.0: Use *canvasapi.submission.Submission.edit()* instead.

Calls PUT /api/v1/sections/:section_id/assignments/:assignment_id/submissions/:user_id

Parameters

- **assignment** (*canvasapi.assignment.Assignment* or int) – The object or ID of the assignment.
- **user** (*canvasapi.user.User* or int or str) – The object or ID of the user.

Return type *canvasapi.submission.Submission*

6.61 Submission

class canvasapi.submission.**Submission**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

create_submission_peer_review(*user*, ***kwargs*)

Create a peer review for this submission.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:submission_id/peer_reviews

Parameters **user** (*canvasapi.user.User* or int) – The user object or ID to retrieve notifications for.

Return type `canvasapi.peer_review.PeerReview`

`delete_submission_peer_review`(*user*, ***kwargs*)
Delete a peer review for this submission.

Calls DELETE /api/v1/courses/:course_id/assignments/:assignment_id/ submissions/:submission_id/peer_reviews

Parameters **`user`** (*`canvasapi.user.User`* or int) – The user object or ID to retrieve notifications for.

Return type `canvasapi.peer_review.PeerReview`

`edit`(***kwargs*)
Comment on and/or update the grading for a student's assignment submission.

Calls PUT /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id

Return type *`canvasapi.submission.Submission`*

`get_submission_peer_reviews`(***kwargs*)
Get a list of all Peer Reviews this submission.

Calls GET /api/v1/courses/:course_id/assignments/:assignment_id/ submissions/:submission_id/peer_reviews

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.peer_review.PeerReview`

`mark_read`(***kwargs*)
Mark submission as read. No request fields are necessary.

Calls PUT /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id/read

Returns True if successfully marked as read.

Return type bool

`mark_unread`(***kwargs*)
Mark submission as unread. No request fields are necessary.

Calls DELETE /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id/read

Returns True if successfully marked as unread.

Return type bool

`upload_comment`(*file*, ***kwargs*)
Upload a file to attach to this submission as a comment.

Calls POST /api/v1/courses/:course_id/assignments/:assignment_id/submissions/:user_id/comments/files

Parameters **`file`** (*`file or str`*) – The file or path of the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.62 Tab

`class` `canvasapi.tab.`**`Tab`**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

update (***kwargs*)

Update a tab for a course.

Note: Home and Settings tabs are not manageable, and can't be hidden or moved.

Calls PUT /api/v1/courses/:course_id/tabs/:tab_id

Return type *canvasapi.tab.Tab*

6.63 Uploader

class canvasapi.upload.**Uploader** (*requester*, *url*, *file*, ***kwargs*)

Upload a file to Canvas.

Parameters

- **requester** (canvasapi.requester.Requester) – The canvasapi.requester.Requester to pass requests through.
- **url** (*str*) – The URL to upload the file to.
- **file** (*file or str*) – A file handler or path of the file to upload.

request_upload_token (*file*)

Request an upload token.

Parameters **file** – A file handler pointing to the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

start ()

Kick off uploading process. Handles open/closing file if a path is passed.

Calls request_upload_token

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

upload (*response*, *file*)

Upload the file.

Parameters

- **response** (*dict*) – The response from the upload request.
- **file** – A file handler pointing to the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

6.64 User

class canvasapi.user.**User**(*requester*, *attributes*)

Parameters

- **requester** (canvasapi.requester.Requester) – The requester to pass HTTP requests through.
- **attributes** (*dict*) – The JSON object to build this object with.

add_observee(*observee_id*)

Registers a user as being observed by the given user.

Calls PUT /api/v1/users/:user_id/observees/:observee_id

Parameters **observee_id** (*int*) – The login id for the user to observe.

Return type

class *canvasapi.user.User*

add_observee_with_credentials(***kwargs*)

Register the given user to observe another user, given the observee's credentials.

Calls POST /api/v1/users/:user_id/observees

Return type *canvasapi.user.User*

create_communication_channel(***kwargs*)

Create a communication channel for this user

Calls POST /api/v1/users/:user_id/communication_channels

Return type *canvasapi.communication_channel.CommunicationChannel*

create_content_migration(*migration_type*, ***kwargs*)

Create a content migration.

Calls POST /api/v1/users/:user_id/content_migrations

Parameters **migration_type** (str or *canvasapi.content_migration.Migrator*) – The migrator type to use in this migration

Return type *canvasapi.content_migration.ContentMigration*

create_folder(*name*, ***kwargs*)

Creates a folder in this user.

Calls POST /api/v1/users/:user_id/folders

Parameters **name** (*str*) – The name of the folder.

Return type *canvasapi.folder.Folder*

edit(***kwargs*)

Modify this user's information.

Calls PUT /api/v1/users/:id

Return type *canvasapi.user.User*

export_content(*export_type*, ***kwargs*)

Begin a content export job for a user.

Calls POST /api/v1/users/:user_id/content_exports

Parameters **`export_type`** (*str*) – The type of content to export.

Return type *`canvasapi.content_export.ContentExport`*

`get_assignments`(*course*, ***kwargs*)

Return the list of assignments for this user if the current user (the API key owner) has rights to view. See List assignments for valid arguments.

Calls GET /api/v1/users/:user_id/courses/:course_id/assignments

Parameters **`course`** (*`canvasapi.course.Course`* or int) – The object or ID of the course to retrieve.

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.assignment.Assignment`*

`get_avatars`()

Retrieve the possible user avatar options that can be set with the user update endpoint.

Calls GET /api/v1/users/:user_id/avatars

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.avatar.Avatar`*

`get_calendar_events_for_user`(***kwargs*)

List calendar events that the current user can view or manage.

Calls GET /api/v1/users/:user_id/calendar_events

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.calendar_event.CalendarEvent`*

`get_closed_poll_sessions`(***kwargs*)

Returns a paginated list of all closed poll sessions available to the current user.

Calls GET /api/v1/poll_sessions/closed

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.poll_session.PollSession`*

`get_color`(*asset_string*)

Return the custom colors that have been saved by this user for a given context.

The *asset_string* parameter should be in the format ‘context_id’, for example ‘course_42’.

Calls GET /api/v1/users/:id/colors/:asset_string

Parameters **`asset_string`** (*str*) – The asset to retrieve the color from.

Return type dict

`get_colors`()

Return all custom colors that have been saved by this user.

Calls GET /api/v1/users/:id/colors

Return type dict

`get_communication_channels`(***kwargs*)

List communication channels for the specified user, sorted by position.

Calls GET /api/v1/users/:user_id/communication_channels

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.communication_channel.CommunicationChannel`*

get_content_export(*content_export*, ***kwargs*)

Return information about a single content export.

Calls GET /api/v1/users/:user_id/content_exports/:id

Parameters **content_export** (int or *canvasapi.content_export.ContentExport*) – The object or ID of the content export to show.

Return type *canvasapi.content_export.ContentExport*

get_content_exports(***kwargs*)

Return a paginated list of the past and pending content export jobs for a user.

Calls GET /api/v1/users/:user_id/content_exports

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_export.ContentExport*

get_content_migration(*content_migration*, ***kwargs*)

Retrive a content migration by its ID

Calls GET /api/v1/users/:user_id/content_migrations/:id

Parameters **content_migration** (int, str or *canvasapi.content_migration.ContentMigration*) – The object or ID of the content migration to retrieve.

Return type *canvasapi.content_migration.ContentMigration*

get_content_migrations(***kwargs*)

List content migrations that the current account can view or manage.

Calls GET /api/v1/users/:user_id/content_migrations/

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_migration.ContentMigration*

get_courses(***kwargs*)

Retrieve all courses this user is enrolled in.

Calls GET /api/v1/users/:user_id/courses

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.course.Course*

get_enrollments(***kwargs*)

List all of the enrollments for this user.

Calls GET /api/v1/users/:user_id/enrollments

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.enrollment.Enrollment*

get_file(*file*, ***kwargs*)

Return the standard attachment json object for a file.

Calls GET /api/v1/users/:user_id/files/:id

Parameters **file** (*canvasapi.file.File* or int) – The object or ID of the file to retrieve.

Return type *canvasapi.file.File*

get_files(***kwargs*)

Returns the paginated list of files for the user.

Calls GET /api/v1/users/:user_id/files

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.file.File*

get_folder(*folder*)

Returns the details for a user's folder

Calls GET /api/v1/users/:user_id/folders/:id

Parameters **folder** (*canvasapi.folder.Folder* or int) – The object or ID of the folder to retrieve.

Return type *canvasapi.folder.Folder*

get_folders(***kwargs*)

Returns the paginated list of all folders for the given user. This will be returned as a flat list containing all subfolders as well.

Calls GET /api/v1/users/:user_id/folders

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.folder.Folder*

get_migration_systems(***kwargs*)

Return a list of migration systems.

Calls GET /api/v1/users/:user_id/content_migrations/migrators

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.content_migration.Migrator*

get_missing_submissions()

Retrieve all past-due assignments for which the student does not have a submission.

Calls GET /api/v1/users/:user_id/missing_submissions

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.assignment.Assignment*

get_observees(***kwargs*)

List the users that the given user is observing

Calls GET /api/v1/users/:user_id/observees

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.user.User*

get_open_poll_sessions(***kwargs*)

Returns a paginated list of all opened poll sessions available to the current user.

Calls GET /api/v1/poll_sessions/opened

Return type `canvasapi.paginated_list.PaginatedList` of *canvasapi.poll_session.PollSession*

get_page_views(***kwargs*)

Retrieve this user's page views.

Calls GET /api/v1/users/:user_id/page_views

Return type `canvasapi.paginated_list.PaginatedList` of `canvasapi.course.PageView`

get_profile(***kwargs*)

Retrieve this user's profile.

Calls GET /api/v1/users/:user_id/profile

Return type dict

get_user_logins(***kwargs*)

Given a user ID, return that user's logins for the given account.

Calls GET /api/v1/users/:user_id/logins

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.login.Login`*

list_calendar_events_for_user(***kwargs*)

List calendar events that the current user can view or manage.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_calendar_events_for_user()`* instead.

Calls GET /api/v1/users/:user_id/calendar_events

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.calendar_event.CalendarEvent`*

list_communication_channels(***kwargs*)

List communication channels for the specified user, sorted by position.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_communication_channels()`* instead.

Calls GET /api/v1/users/:user_id/communication_channels

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.communication_channel.CommunicationChannel`*

list_files(***kwargs*)

Returns the paginated list of files for the user.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_files()`* instead.

Calls GET /api/v1/users/:user_id/files

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.file.File`*

list_folders(***kwargs*)

Returns the paginated list of all folders for the given user. This will be returned as a flat list containing all subfolders as well.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_folders()`* instead.

Calls GET /api/v1/users/:user_id/folders

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.folder.Folder`*

`list_observees`(***kwargs*)
List the users that the given user is observing

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_observees()`* instead.

Calls GET /api/v1/users/:user_id/observees

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.user.User`*

`list_user_logins`(***kwargs*)
Given a user ID, return that user's logins for the given account.

> **Warning:** Deprecated since version 0.10.0: Use *`canvasapi.user.User.get_user_logins()`* instead.

Calls GET /api/v1/users/:user_id/logins

Return type `canvasapi.paginated_list.PaginatedList` of *`canvasapi.login.Login`*

`merge_into`(*destination_user*)
Merge this user into another user.

Calls PUT /api/v1/users/:id/merge_into/:destination_user_id

Parameters **`destination_user`** (*`canvasapi.user.User`* or int) – The object or ID of the user to merge into.

Return type *`canvasapi.user.User`*

`remove_observee`(*observee_id*)
Unregisters a user as being observed by the given user.

Calls DELETE /api/v1/users/:user_id/observees/:observee_id

Parameters **`observee_id`** (*`int`*) – The login id for the user to observe.

Return type

class *canvasapi.user.User*

`show_observee`(*observee_id*)
Gets information about an observed user.

Calls GET /api/v1/users/:user_id/observees/:observee_id

Parameters **`observee_id`** (*`int`*) – The login id for the user to observe.

Return type

class *canvasapi.user.User*

update_color(*asset_string*, *hexcode*)

Update a custom color for this user for a given context.

This allows colors for the calendar and elsewhere to be customized on a user basis.

The *asset_string* parameter should be in the format 'context_id', for example 'course_42'. The *hexcode* parameter need not include the '#'.

Calls PUT /api/v1/users/:id/colors/:asset_string

Parameters

- **asset_string** (*str*) – The asset to modify the color for.
- **hexcode** (*str*) – The hexcode of the color to use.

Return type dict

update_settings(***kwargs*)

Update this user's settings.

Calls PUT /api/v1/users/:id/settings

Return type dict

upload(*file*, ***kwargs*)

Upload a file for a user.

NOTE: You *must* have authenticated with this user's API key to upload on their behalf no matter what permissions the issuer of the request has.

Calls POST /api/v1/users/:user_id/files

Parameters **file** (*file or str*) – The file or path of the file to upload.

Returns True if the file uploaded successfully, False otherwise, and the JSON response from the API.

Return type tuple

Index