

---

# **canvasapi Documentation**

***Release 0.11.0***

**University of Central Florida (Center for Distributed Learning)**

**Mar 04, 2019**



---

## Contents

---

<b>1</b>	<b>Getting Started with CanvasAPI</b>	<b>3</b>
1.1	Installing CanvasAPI . . . . .	3
1.2	Usage . . . . .	3
1.3	Working with Canvas Objects . . . . .	3
<b>2</b>	<b>Examples</b>	<b>7</b>
2.1	Boilerplate . . . . .	7
2.2	Accounts . . . . .	7
2.3	Users . . . . .	8
2.4	Logins . . . . .	9
2.5	Courses . . . . .	9
2.6	Assignments . . . . .	11
<b>3</b>	<b>Keyword Arguments</b>	<b>15</b>
3.1	Basic Parameters . . . . .	15
3.2	List Parameters . . . . .	15
3.3	Nested Parameters . . . . .	16
3.4	Complex Parameters . . . . .	17
<b>4</b>	<b>Class Reference</b>	<b>19</b>
4.1	Canvas . . . . .	19
4.2	Account . . . . .	27
4.3	AccountNotification . . . . .	37
4.4	AccountReport . . . . .	37
4.5	Role . . . . .	37
4.6	SSOSettings . . . . .	37
4.7	AppointmentGroup . . . . .	38
4.8	Assignment . . . . .	38
4.9	AssignmentGroup . . . . .	40
4.10	AuthenticationProvider . . . . .	40
4.11	Avatar . . . . .	41
4.12	BlueprintMigration . . . . .	41
4.13	BlueprintSubscription . . . . .	41
4.14	BlueprintTemplate . . . . .	42
4.15	ChangeRecord . . . . .	43
4.16	Bookmark . . . . .	43
4.17	CalendarEvent . . . . .	44

4.18	CommunicationChannel . . . . .	44
4.19	ContentMigration . . . . .	46
4.20	Conversation . . . . .	47
4.21	Course . . . . .	48
4.22	CourseNickname . . . . .	64
4.23	CurrentUser . . . . .	65
4.24	DiscussionEntry . . . . .	66
4.25	DiscussionTopic . . . . .	67
4.26	Enrollment . . . . .	70
4.27	EnrollmentTerm . . . . .	70
4.28	ExternalTool . . . . .	71
4.29	File . . . . .	72
4.30	Folder . . . . .	72
4.31	Group . . . . .	74
4.32	GroupMembership . . . . .	80
4.33	GroupCategory . . . . .	81
4.34	Login . . . . .	82
4.35	Module . . . . .	82
4.36	ModuleItem . . . . .	84
4.37	Outcome . . . . .	84
4.38	OutcomeLink . . . . .	85
4.39	OutcomeGroup . . . . .	85
4.40	Page . . . . .	88
4.41	PageRevision . . . . .	89
4.42	Progress . . . . .	89
4.43	Quiz . . . . .	90
4.44	QuizExtension . . . . .	92
4.45	QuizQuestion . . . . .	92
4.46	QuizSubmission . . . . .	92
4.47	QuizGroup . . . . .	93
4.48	Rubric . . . . .	94
4.49	Section . . . . .	94
4.50	Submission . . . . .	97
4.51	Tab . . . . .	98
4.52	Uploader . . . . .	98
4.53	User . . . . .	99

Contents:



---

## 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. Canvas-API 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:

```
# 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

```
# 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()
```



---

## 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']
    }
)
```

## 4.1 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`

**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\_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\_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*

## 4.2 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.

**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 longer 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 notification 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\_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\_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\_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\_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\_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\_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`

**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`

## 4.3 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.

## 4.4 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.

## 4.5 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.

## 4.6 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.

## 4.7 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*

## 4.8 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\_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

## 4.9 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`

## 4.10 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`

## 4.11 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.

## 4.12 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`

## 4.13 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`

## 4.14 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 successfully 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

## 4.15 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.

## 4.16 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

## 4.17 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

## 4.18 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.

**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:** Deprecatd 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:** Deprecatd 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`

## 4.19 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.

## 4.20 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*

## 4.21 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\_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`

**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\_migration** (content\_migration, \*\*kwargs)

Retrieve 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\_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\_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\_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*

**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:** Deprecatd 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
...     }
... ])
```

(continues on next page)

(continued from previous page)

```
...     },
...     {
...         '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:** Deprecatcd 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:** Deprecatcd 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`

## 4.22 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`

## 4.23 CurrentUser

**class** `canvasapi.current_user.CurrentUser` (*\_requester*)

**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\_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`

## 4.24 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`

## 4.25 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*)

Retrieve 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)

Retrieve 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*

## 4.26 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*

## 4.27 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`

## 4.28 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

## 4.29 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`

## 4.30 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

## 4.31 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 url of 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`

**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\_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`

## 4.32 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*

## 4.33 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`

## 4.34 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`

## 4.35 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`

## 4.36 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

## 4.37 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



## 4.38 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

## 4.39 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

## 4.40 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`

## 4.41 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

## 4.42 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`

## 4.43 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.

**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`

**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
...     },
...     {
```

(continues on next page)

(continued from previous page)

```
...         'user_id': 2,
...         'extra_attempts': 3
...     },
...     {
...         'user_id': 3,
...         'extra_time': 20
...     }
... ])
```

## 4.44 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.

## 4.45 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*

## 4.46 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.

**complete** (*\*\*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`

**Return type** `canvasapi.quiz.QuizSubmission`

**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`

## 4.47 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 receive 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`

## 4.48 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.

## 4.49 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`

## 4.50 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.

**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`

**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

## 4.51 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`

## 4.52 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

## 4.53 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\_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`

**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\_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\_migration** (*content\_migration, \*\*kwargs*)  
 Retrieve 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\_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\_observes** (\*\*kwargs)

List the users that the given user is observing

**Warning:** Deprecated since version 0.10.0: Use `canvasapi.user.User.get_observes()` 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



## A

Account (*class in canvasapi.account*), 27  
AccountNotification (*class in canvasapi.account*), 37  
AccountReport (*class in canvasapi.account*), 37  
activate\_role() (*canvasapi.account.Account method*), 27  
add\_authentication\_providers() (*canvasapi.account.Account method*), 27  
add\_grading\_standards() (*canvasapi.account.Account method*), 27  
add\_grading\_standards() (*canvasapi.course.Course method*), 48  
add\_message() (*canvasapi.conversation.Conversation method*), 47  
add\_observee() (*canvasapi.user.User method*), 99  
add\_observee\_with\_credentials() (*canvasapi.user.User method*), 99  
add\_recipients() (*canvasapi.conversation.Conversation method*), 47  
AppointmentGroup (*class in canvasapi.appointment\_group*), 38  
assign\_members() (*canvasapi.group.GroupCategory method*), 81  
Assignment (*class in canvasapi.assignment*), 38  
AssignmentGroup (*class in canvasapi.assignment*), 40  
associated\_course\_migration() (*canvasapi.blueprint.BlueprintTemplate method*), 42  
AuthenticationProvider (*class in canvasapi.authentication\_provider*), 40  
Avatar (*class in canvasapi.avatar*), 41

## B

BlueprintMigration (*class in canvasapi.blueprint*), 41

BlueprintSubscription (*class in canvasapi.blueprint*), 41  
BlueprintTemplate (*class in canvasapi.blueprint*), 42  
Bookmark (*class in canvasapi.bookmark*), 43

## C

CalendarEvent (*class in canvasapi.calendar\_event*), 44  
Canvas (*class in canvasapi*), 19  
change\_blueprint\_restrictions() (*canvasapi.blueprint.BlueprintTemplate method*), 42  
ChangeRecord (*class in canvasapi.blueprint*), 43  
clear\_course\_nicknames() (*canvasapi.Canvas method*), 19  
close\_notification\_for\_user() (*canvasapi.account.Account method*), 27  
CommunicationChannel (*class in canvasapi.communication\_channel*), 44  
complete() (*canvasapi.module.ModuleItem method*), 84  
complete() (*canvasapi.quiz.QuizSubmission method*), 93  
conclude() (*canvasapi.course.Course method*), 48  
ContentMigration (*class in canvasapi.content\_migration*), 46  
Conversation (*class in canvasapi.conversation*), 47  
conversations\_batch\_update() (*canvasapi.Canvas method*), 19  
conversations\_get\_running\_batches() (*canvasapi.Canvas method*), 19  
conversations\_mark\_all\_as\_read() (*canvasapi.Canvas method*), 20  
conversations\_unread\_count() (*canvasapi.Canvas method*), 20  
copy\_file() (*canvasapi.folder.Folder method*), 72  
Course (*class in canvasapi.course*), 48  
CourseNickname (*class in canvasapi.course*), 64

[create\\_account\(\)](#) (*canvasapi.account.Account method*), 27  
[create\\_account\(\)](#) (*canvasapi.Canvas method*), 20  
[create\\_appointment\\_group\(\)](#) (*canvasapi.Canvas method*), 20  
[create\\_assignment\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_assignment\\_group\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_assignment\\_overrides\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_bookmark\(\)](#) (*canvasapi.current\_user.CurrentUser method*), 65  
[create\\_calendar\\_event\(\)](#) (*canvasapi.Canvas method*), 20  
[create\\_content\\_migration\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_content\\_migration\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_content\\_migration\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_content\\_migration\(\)](#) (*canvasapi.user.User method*), 99  
[create\\_conversation\(\)](#) (*canvasapi.Canvas method*), 20  
[create\\_course\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_course\\_section\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_discussion\\_topic\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_discussion\\_topic\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_enrollment\\_term\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_external\\_feed\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_external\\_feed\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_external\\_tool\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_external\\_tool\(\)](#) (*canvasapi.course.Course method*), 49  
[create\\_folder\(\)](#) (*canvasapi.course.Course method*), 50  
[create\\_folder\(\)](#) (*canvasapi.folder.Folder method*), 72  
[create\\_folder\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_folder\(\)](#) (*canvasapi.user.User method*), 100  
[create\\_group\(\)](#) (*canvasapi.Canvas method*), 20  
[create\\_group\(\)](#) (*canvasapi.group.GroupCategory method*), 81  
[create\\_group\\_category\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_group\\_category\(\)](#) (*canvasapi.course.Course method*), 50  
[create\\_membership\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_module\(\)](#) (*canvasapi.course.Course method*), 50  
[create\\_module\\_item\(\)](#) (*canvasapi.module.Module method*), 82  
[create\\_notification\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_override\(\)](#) (*canvasapi.assignment.Assignment method*), 38  
[create\\_page\(\)](#) (*canvasapi.course.Course method*), 50  
[create\\_page\(\)](#) (*canvasapi.group.Group method*), 74  
[create\\_question\(\)](#) (*canvasapi.quiz.Quiz method*), 90  
[create\\_question\\_group\(\)](#) (*canvasapi.quiz.Quiz method*), 90  
[create\\_quiz\(\)](#) (*canvasapi.course.Course method*), 50  
[create\\_role\(\)](#) (*canvasapi.account.Account method*), 28  
[create\\_subaccount\(\)](#) (*canvasapi.account.Account method*), 29  
[create\\_subgroup\(\)](#) (*canvasapi.outcome.OutcomeGroup method*), 85  
[create\\_submission\(\)](#) (*canvasapi.quiz.Quiz method*), 90  
[create\\_user\(\)](#) (*canvasapi.account.Account method*), 29  
[create\\_user\\_login\(\)](#) (*canvasapi.account.Account method*), 29  
[cross\\_list\\_section\(\)](#) (*canvasapi.section.Section method*), 94  
[CurrentUser](#) (*class in canvasapi.current\_user*), 65

## D

[deactivate\(\)](#) (*canvasapi.enrollment.Enrollment method*), 70  
[deactivate\\_role\(\)](#) (*canvasapi.account.Account method*), 29  
[decross\\_list\\_section\(\)](#) (*canvasapi.section.Section method*), 94  
[delete\(\)](#) (*canvasapi.account.Account method*), 29  
[delete\(\)](#) (*canvasapi.appointment\_group.AppointmentGroup method*), 38  
[delete\(\)](#) (*canvasapi.assignment.Assignment method*), 38  
[delete\(\)](#) (*canvasapi.assignment.AssignmentGroup method*), 40



delete() (*canvasapi.authentication\_provider.AuthenticationProvider method*), 40

delete() (*canvasapi.bookmark.Bookmark method*), 43

delete() (*canvasapi.calendar\_event.CalendarEvent method*), 44

delete() (*canvasapi.conversation.Conversation method*), 48

delete() (*canvasapi.course.Course method*), 50

delete() (*canvasapi.discussion\_topic.DiscussionEntry method*), 66

delete() (*canvasapi.discussion\_topic.DiscussionTopic method*), 68

delete() (*canvasapi.enrollment\_term.EnrollmentTerm method*), 70

delete() (*canvasapi.external\_tool.ExternalTool method*), 71

delete() (*canvasapi.file.File method*), 72

delete() (*canvasapi.folder.Folder method*), 72

delete() (*canvasapi.group.Group method*), 74

delete() (*canvasapi.group.GroupCategory method*), 81

delete() (*canvasapi.login.Login method*), 82

delete() (*canvasapi.module.Module method*), 83

delete() (*canvasapi.module.ModuleItem method*), 84

delete() (*canvasapi.outcome.OutcomeGroup method*), 85

delete() (*canvasapi.page.Page method*), 88

delete() (*canvasapi.quiz.Quiz method*), 90

delete() (*canvasapi.quiz.QuizGroup method*), 93

delete() (*canvasapi.quiz.QuizQuestion method*), 92

delete() (*canvasapi.section.Section method*), 94

delete\_external\_feed() (*canvasapi.course.Course method*), 51

delete\_external\_feed() (*canvasapi.group.Group method*), 75

delete\_messages() (*canvasapi.conversation.Conversation method*), 48

delete\_user() (*canvasapi.account.Account method*), 29

DiscussionEntry (class in *canvasapi.discussion\_topic*), 66

DiscussionTopic (class in *canvasapi.discussion\_topic*), 67

download() (*canvasapi.file.File method*), 72

delete() (*canvasapi.authentication\_provider.AuthenticationProvider method*), 44

edit() (*canvasapi.conversation.Conversation method*), 48

edit() (*canvasapi.enrollment\_term.EnrollmentTerm method*), 71

edit() (*canvasapi.external\_tool.ExternalTool method*), 71

edit() (*canvasapi.group.Group method*), 75

edit() (*canvasapi.login.Login method*), 82

edit() (*canvasapi.module.Module method*), 83

edit() (*canvasapi.module.ModuleItem method*), 84

edit() (*canvasapi.page.Page method*), 88

edit() (*canvasapi.quiz.Quiz method*), 90

edit() (*canvasapi.quiz.QuizQuestion method*), 92

edit() (*canvasapi.section.Section method*), 95

edit() (*canvasapi.submission.Submission method*), 97

edit() (*canvasapi.user.User method*), 100

edit\_front\_page() (*canvasapi.course.Course method*), 51

edit\_front\_page() (*canvasapi.group.Group method*), 75

enroll\_user() (*canvasapi.course.Course method*), 51

Enrollment (class in *canvasapi.enrollment*), 70

EnrollmentTerm (class in *canvasapi.enrollment\_term*), 70

ExternalTool (class in *canvasapi.external\_tool*), 71

## F

File (class in *canvasapi.file*), 72

Folder (class in *canvasapi.folder*), 72

## G

get\_account() (*canvasapi.Canvas method*), 21

get\_accounts() (*canvasapi.Canvas method*), 21

get\_activity\_stream\_summary() (*canvasapi.Canvas method*), 21

get\_activity\_stream\_summary() (*canvasapi.group.Group method*), 75

get\_admins() (*canvasapi.account.Account method*), 30

get\_all\_outcome\_links\_in\_context() (*canvasapi.account.Account method*), 30

get\_all\_outcome\_links\_in\_context() (*canvasapi.course.Course method*), 51

get\_all\_quiz\_submissions() (*canvasapi.quiz.Quiz method*), 91

get\_announcements() (*canvasapi.Canvas method*), 21

get\_appointment\_group() (*canvasapi.Canvas method*), 21

get\_appointment\_groups() (*canvasapi.Canvas method*), 21

## E

edit() (*canvasapi.appointment\_group.AppointmentGroup method*), 38

edit() (*canvasapi.assignment.Assignment method*), 38

edit() (*canvasapi.assignment.AssignmentGroup method*), 40

edit() (*canvasapi.bookmark.Bookmark method*), 43

<code>get_assignment()</code>	( <i>canvasapi.course.Course method</i> ), 51	<code>get_content_migrations()</code>	( <i>canvasapi.user.User method</i> ), 101
<code>get_assignment_group()</code>	( <i>canvasapi.course.Course method</i> ), 51	<code>get_contents()</code>	( <i>canvasapi.file.File method</i> ), 72
<code>get_assignment_groups()</code>	( <i>canvasapi.course.Course method</i> ), 51	<code>get_conversation()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_assignment_override()</code>	( <i>canvasapi.group.Group method</i> ), 75	<code>get_conversations()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_assignment_override()</code>	( <i>canvasapi.section.Section method</i> ), 95	<code>get_course()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_assignment_overrides()</code>	( <i>canvasapi.course.Course method</i> ), 52	<code>get_course_accounts()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_assignments()</code>	( <i>canvasapi.course.Course method</i> ), 52	<code>get_course_level_assignment_data()</code>	( <i>canvasapi.course.Course method</i> ), 52
<code>get_assignments()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_course_level_participation_data()</code>	( <i>canvasapi.course.Course method</i> ), 52
<code>get_associated_courses()</code>	( <i>canvasapi.blueprint.BlueprintTemplate method</i> ), 42	<code>get_course_level_student_summary_data()</code>	( <i>canvasapi.course.Course method</i> ), 53
<code>get_authentication_provider()</code>	( <i>canvasapi.account.Account method</i> ), 30	<code>get_course_nickname()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_authentication_providers()</code>	( <i>canvasapi.account.Account method</i> ), 30	<code>get_course_nicknames()</code>	( <i>canvasapi.Canvas method</i> ), 22
<code>get_avatars()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_courses()</code>	( <i>canvasapi.account.Account method</i> ), 30
<code>get_blueprint()</code>	( <i>canvasapi.course.Course method</i> ), 52	<code>get_courses()</code>	( <i>canvasapi.Canvas method</i> ), 23
<code>get_bookmark()</code>	( <i>canvasapi.current_user.CurrentUser method</i> ), 65	<code>get_courses()</code>	( <i>canvasapi.user.User method</i> ), 101
<code>get_bookmarks()</code>	( <i>canvasapi.current_user.CurrentUser method</i> ), 65	<code>get_department_level_grade_data_completed()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_calendar_event()</code>	( <i>canvasapi.Canvas method</i> ), 21	<code>get_department_level_grade_data_current()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_calendar_events()</code>	( <i>canvasapi.Canvas method</i> ), 22	<code>get_department_level_grade_data_with_given_term()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_calendar_events_for_user()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_department_level_participation_data_completed()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_color()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_department_level_participation_data_current()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_colors()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_department_level_participation_data_with_given_term()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_communication_channels()</code>	( <i>canvasapi.user.User method</i> ), 100	<code>get_department_level_statistics_completed()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_content_migration()</code>	( <i>canvasapi.account.Account method</i> ), 30	<code>get_department_level_statistics_current()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_content_migration()</code>	( <i>canvasapi.course.Course method</i> ), 52	<code>get_department_level_statistics_with_given_term()</code>	( <i>canvasapi.account.Account method</i> ), 31
<code>get_content_migration()</code>	( <i>canvasapi.group.Group method</i> ), 75	<code>get_details()</code>	( <i>canvasapi.blueprint.BlueprintMigration method</i> ), 41
<code>get_content_migration()</code>	( <i>canvasapi.user.User method</i> ), 101	<code>get_discussion()</code>	( <i>canvasapi.discussion_topic.DiscussionEntry method</i> ), 66
<code>get_content_migrations()</code>	( <i>canvasapi.account.Account method</i> ), 30	<code>get_discussion_topic()</code>	( <i>canvasapi.course.Course method</i> ), 53
<code>get_content_migrations()</code>	( <i>canvasapi.course.Course method</i> ), 52	<code>get_discussion_topic()</code>	( <i>canvasapi.group.Group method</i> ), 75
<code>get_content_migrations()</code>	( <i>canvasapi.group.Group method</i> ), 75	<code>get_discussion_topics()</code>	( <i>canvasapi.course.Course method</i> ), 53

[get\\_discussion\\_topics\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_enrollment\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_enrollment\\_terms\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_enrollments\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_enrollments\(\)](#) (*canvasapi.section.Section method*), 95  
[get\\_enrollments\(\)](#) (*canvasapi.user.User method*), 101  
[get\\_entries\(\)](#) (*canvasapi.discussion\_topic.DiscussionTopic method*), 68  
[get\\_external\\_feeds\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_external\\_feeds\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_external\\_tool\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_external\\_tool\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_external\\_tools\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_external\\_tools\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_file\(\)](#) (*canvasapi.Canvas method*), 23  
[get\\_file\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_file\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_file\(\)](#) (*canvasapi.user.User method*), 101  
[get\\_files\(\)](#) (*canvasapi.course.Course method*), 53  
[get\\_files\(\)](#) (*canvasapi.folder.Folder method*), 73  
[get\\_files\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_files\(\)](#) (*canvasapi.user.User method*), 101  
[get\\_folder\(\)](#) (*canvasapi.Canvas method*), 23  
[get\\_folder\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_folder\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_folder\(\)](#) (*canvasapi.user.User method*), 101  
[get\\_folders\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_folders\(\)](#) (*canvasapi.folder.Folder method*), 73  
[get\\_folders\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_folders\(\)](#) (*canvasapi.user.User method*), 102  
[get\\_full\\_discussion\\_topic\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_full\\_discussion\\_topic\(\)](#) (*canvasapi.group.Group method*), 76  
[get\\_gradeable\\_students\(\)](#) (*canvasapi.assignment.Assignment method*), 38  
[get\\_grading\\_standards\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_grading\\_standards\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_group\(\)](#) (*canvasapi.Canvas method*), 23  
[get\\_group\\_categories\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_group\\_categories\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_group\\_category\(\)](#) (*canvasapi.Canvas method*), 23  
[get\\_group\\_participants\(\)](#) (*canvasapi.Canvas method*), 23  
[get\\_groups\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_groups\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_groups\(\)](#) (*canvasapi.current\_user.CurrentUser method*), 65  
[get\\_groups\(\)](#) (*canvasapi.group.GroupCategory method*), 81  
[get\\_import\\_details\(\)](#) (*canvasapi.blueprint.BlueprintMigration method*), 41  
[get\\_index\\_of\\_reports\(\)](#) (*canvasapi.account.Account method*), 32  
[get\\_linked\\_outcomes\(\)](#) (*canvasapi.outcome.OutcomeGroup method*), 86  
[get\\_membership\(\)](#) (*canvasapi.group.Group method*), 77  
[get\\_memberships\(\)](#) (*canvasapi.group.Group method*), 77  
[get\\_migration\\_issue\(\)](#) (*canvasapi.content\_migration.ContentMigration method*), 46  
[get\\_migration\\_issues\(\)](#) (*canvasapi.content\_migration.ContentMigration method*), 46  
[get\\_migration\\_systems\(\)](#) (*canvasapi.account.Account method*), 33  
[get\\_migration\\_systems\(\)](#) (*canvasapi.course.Course method*), 54  
[get\\_migration\\_systems\(\)](#) (*canvasapi.group.Group method*), 77  
[get\\_migration\\_systems\(\)](#) (*canvasapi.user.User method*), 102  
[get\\_missing\\_submissions\(\)](#) (*canvasapi.user.User method*), 102  
[get\\_module\(\)](#) (*canvasapi.course.Course method*), 55  
[get\\_module\\_item\(\)](#) (*canvasapi.module.Module method*), 83  
[get\\_module\\_items\(\)](#) (*canvasapi.module.Module method*), 83  
[get\\_modules\(\)](#) (*canvasapi.course.Course method*), 55  
[get\\_multiple\\_submissions\(\)](#) (*canvasapi.course.Course method*), 55  
[get\\_multiple\\_submissions\(\)](#) (*canvasapi.section.Section method*), 95

[get\\_observees\(\)](#) (*canvasapi.user.User* method), 102  
[get\\_outcome\(\)](#) (*canvasapi.Canvas* method), 23  
[get\\_outcome\(\)](#) (*canvasapi.outcome.OutcomeLink* method), 85  
[get\\_outcome\\_group\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_outcome\\_group\(\)](#) (*canvasapi.Canvas* method), 24  
[get\\_outcome\\_group\(\)](#) (*canvasapi.course.Course* method), 55  
[get\\_outcome\\_group\(\)](#) (*canvasapi.outcome.OutcomeLink* method), 85  
[get\\_outcome\\_groups\\_in\\_context\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_outcome\\_groups\\_in\\_context\(\)](#) (*canvasapi.course.Course* method), 55  
[get\\_outcome\\_result\\_rollups\(\)](#) (*canvasapi.course.Course* method), 55  
[get\\_outcome\\_results\(\)](#) (*canvasapi.course.Course* method), 55  
[get\\_override\(\)](#) (*canvasapi.assignment.Assignment* method), 39  
[get\\_overrides\(\)](#) (*canvasapi.assignment.Assignment* method), 39  
[get\\_page\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_page\(\)](#) (*canvasapi.group.Group* method), 77  
[get\\_page\\_views\(\)](#) (*canvasapi.user.User* method), 102  
[get\\_pages\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_pages\(\)](#) (*canvasapi.group.Group* method), 77  
[get\\_parent\(\)](#) (*canvasapi.content\_migration.ContentMigration* method), 46  
[get\\_parent\(\)](#) (*canvasapi.discussion\_topic.DiscussionTopic* method), 68  
[get\\_parent\(\)](#) (*canvasapi.external\_tool.ExternalTool* method), 71  
[get\\_parent\(\)](#) (*canvasapi.page.Page* method), 88  
[get\\_parent\(\)](#) (*canvasapi.page.PageRevision* method), 89  
[get\\_preference\(\)](#) (*canvasapi.communication\_channel.CommunicationChannel* method), 44  
[get\\_preference\\_categories\(\)](#) (*canvasapi.communication\_channel.CommunicationChannel* method), 44  
[get\\_preferences\(\)](#) (*canvasapi.communication\_channel.CommunicationChannel* method), 44  
[get\\_profile\(\)](#) (*canvasapi.user.User* method), 102  
[get\\_progress\(\)](#) (*canvasapi.Canvas* method), 24  
[get\\_progress\(\)](#) (*canvasapi.content\_migration.ContentMigration* method), 46  
[get\\_question\(\)](#) (*canvasapi.quiz.Quiz* method), 91  
[get\\_questions\(\)](#) (*canvasapi.quiz.Quiz* method), 91  
[get\\_quiz\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_quiz\\_group\(\)](#) (*canvasapi.quiz.Quiz* method), 91  
[get\\_quiz\\_submission\(\)](#) (*canvasapi.quiz.Quiz* method), 91  
[get\\_quizzes\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_recent\\_students\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_replies\(\)](#) (*canvasapi.discussion\_topic.DiscussionEntry* method), 66  
[get\\_reports\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_revision\\_by\\_id\(\)](#) (*canvasapi.page.Page* method), 88  
[get\\_revisions\(\)](#) (*canvasapi.page.Page* method), 88  
[get\\_role\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_roles\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_root\\_outcome\\_group\(\)](#) (*canvasapi.account.Account* method), 33  
[get\\_root\\_outcome\\_group\(\)](#) (*canvasapi.Canvas* method), 24  
[get\\_root\\_outcome\\_group\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_rubric\(\)](#) (*canvasapi.account.Account* method), 34  
[get\\_rubric\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_rubrics\(\)](#) (*canvasapi.account.Account* method), 34  
[get\\_rubrics\(\)](#) (*canvasapi.course.Course* method), 56  
[get\\_section\(\)](#) (*canvasapi.Canvas* method), 24  
[get\\_section\(\)](#) (*canvasapi.course.Course* method), 57  
[get\\_sections\(\)](#) (*canvasapi.course.Course* method), 57  
[get\\_sessionless\\_launch\\_url\(\)](#) (*canvasapi.external\_tool.ExternalTool* method), 71  
[get\\_settings\(\)](#) (*canvasapi.course.Course* method), 57  
[get\\_single\\_grading\\_standard\(\)](#) (*canvasapi.account.Account* method), 34  
[get\\_single\\_grading\\_standard\(\)](#) (*canvasapi.course.Course* method), 57  
[get\\_subaccounts\(\)](#) (*canvasapi.account.Account* method), 34  
[get\\_subgroups\(\)](#) (*canvasapi.outcome.OutcomeGroup* method), 86

[get\\_submission\(\)](#) (*canvasapi.assignment.Assignment method*), 39  
[get\\_submission\(\)](#) (*canvasapi.course.Course method*), 57  
[get\\_submission\(\)](#) (*canvasapi.section.Section method*), 95  
[get\\_submissions\(\)](#) (*canvasapi.assignment.Assignment method*), 39  
[get\\_tabs\(\)](#) (*canvasapi.course.Course method*), 57  
[get\\_tabs\(\)](#) (*canvasapi.group.Group method*), 77  
[get\\_times\(\)](#) (*canvasapi.quiz.QuizSubmission method*), 93  
[get\\_todo\\_items\(\)](#) (*canvasapi.Canvas method*), 24  
[get\\_topic\\_entries\(\)](#) (*canvasapi.discussion\_topic.DiscussionTopic method*), 68  
[get\\_unsynced\\_changes\(\)](#) (*canvasapi.blueprint.BlueprintTemplate method*), 42  
[get\\_upcoming\\_events\(\)](#) (*canvasapi.Canvas method*), 24  
[get\\_user\(\)](#) (*canvasapi.Canvas method*), 24  
[get\\_user\(\)](#) (*canvasapi.course.Course method*), 57  
[get\\_user\\_in\\_a\\_course\\_level\\_assignment\\_data\(\)](#) (*canvasapi.course.Course method*), 58  
[get\\_user\\_in\\_a\\_course\\_level\\_messaging\\_data\(\)](#) (*canvasapi.course.Course method*), 58  
[get\\_user\\_in\\_a\\_course\\_level\\_participation\\_data\(\)](#) (*canvasapi.course.Course method*), 58  
[get\\_user\\_logins\(\)](#) (*canvasapi.account.Account method*), 34  
[get\\_user\\_logins\(\)](#) (*canvasapi.user.User method*), 102  
[get\\_user\\_notifications\(\)](#) (*canvasapi.account.Account method*), 34  
[get\\_user\\_participants\(\)](#) (*canvasapi.Canvas method*), 25  
[get\\_users\(\)](#) (*canvasapi.account.Account method*), 34  
[get\\_users\(\)](#) (*canvasapi.course.Course method*), 58  
[get\\_users\(\)](#) (*canvasapi.group.Group method*), 77  
[get\\_users\(\)](#) (*canvasapi.group.GroupCategory method*), 81  
[Group](#) (*class in canvasapi.group*), 74  
[GroupCategory](#) (*class in canvasapi.group*), 81  
[GroupMembership](#) (*class in canvasapi.group*), 80  
**I**  
[import\\_outcome\\_group\(\)](#) (*canvasapi.outcome.OutcomeGroup method*), 86  
[invite\(\)](#) (*canvasapi.group.Group method*), 77  
**L**  
[link\\_existing\(\)](#) (*canvasapi.outcome.OutcomeGroup method*), 86  
[link\\_new\(\)](#) (*canvasapi.outcome.OutcomeGroup method*), 86  
[list\\_appointment\\_groups\(\)](#) (*canvasapi.Canvas method*), 25  
[list\\_assignment\\_groups\(\)](#) (*canvasapi.course.Course method*), 58  
[list\\_authentication\\_providers\(\)](#) (*canvasapi.account.Account method*), 35  
[list\\_blueprint\\_imports\(\)](#) (*canvasapi.blueprint.BlueprintSubscription method*), 41  
[list\\_blueprint\\_migrations\(\)](#) (*canvasapi.blueprint.BlueprintTemplate method*), 42  
[list\\_blueprint\\_subscriptions\(\)](#) (*canvasapi.course.Course method*), 58  
[list\\_bookmarks\(\)](#) (*canvasapi.current\_user.CurrentUser method*), 65  
[list\\_calendar\\_events\(\)](#) (*canvasapi.Canvas method*), 25  
[list\\_calendar\\_events\\_for\\_user\(\)](#) (*canvasapi.user.User method*), 102  
[list\\_communication\\_channels\(\)](#) (*canvasapi.user.User method*), 103  
[list\\_enrollment\\_terms\(\)](#) (*canvasapi.account.Account method*), 35  
[list\\_entries\(\)](#) (*canvasapi.discussion\_topic.DiscussionTopic method*), 68  
[list\\_external\\_feeds\(\)](#) (*canvasapi.course.Course method*), 59  
[list\\_external\\_feeds\(\)](#) (*canvasapi.group.Group method*), 78  
[list\\_files\(\)](#) (*canvasapi.course.Course method*), 59  
[list\\_files\(\)](#) (*canvasapi.folder.Folder method*), 73  
[list\\_files\(\)](#) (*canvasapi.group.Group method*), 78  
[list\\_files\(\)](#) (*canvasapi.user.User method*), 103  
[list\\_folders\(\)](#) (*canvasapi.course.Course method*), 59  
[list\\_folders\(\)](#) (*canvasapi.folder.Folder method*), 73  
[list\\_folders\(\)](#) (*canvasapi.group.Group method*), 78  
[list\\_folders\(\)](#) (*canvasapi.user.User method*), 103  
[list\\_gradeable\\_students\(\)](#) (*canvasapi.course.Course method*), 59  
[list\\_group\\_categories\(\)](#) (*canvasapi.account.Account method*), 35  
[list\\_group\\_categories\(\)](#) (*canvasapi.course.Course method*), 59  
[list\\_group\\_participants\(\)](#) (*canvasapi.Canvas method*), 25  
[list\\_groups\(\)](#) (*canvasapi.account.Account method*), 34



method), 35

list\_groups() (canvasapi.course.Course method), 60

list\_groups() (canvasapi.current\_user.CurrentUser method), 66

list\_groups() (canvasapi.group.GroupCategory method), 81

list\_linked\_outcomes() (canvasapi.outcome.OutcomeGroup method), 87

list\_memberships() (canvasapi.group.Group method), 78

list\_module\_items() (canvasapi.module.Module method), 83

list\_multiple\_submissions() (canvasapi.course.Course method), 60

list\_multiple\_submissions() (canvasapi.section.Section method), 95

list\_observes() (canvasapi.user.User method), 103

list\_preference\_categories() (canvasapi.communication\_channel.CommunicationChannel method), 44

list\_preferences() (canvasapi.communication\_channel.CommunicationChannel method), 45

list\_replies() (canvasapi.discussion\_topic.DiscussionEntry method), 66

list\_revisions() (canvasapi.page.Page method), 88

list\_roles() (canvasapi.account.Account method), 35

list\_rubrics() (canvasapi.account.Account method), 36

list\_rubrics() (canvasapi.course.Course method), 60

list\_sections() (canvasapi.course.Course method), 60

list\_subgroups() (canvasapi.outcome.OutcomeGroup method), 87

list\_submissions() (canvasapi.course.Course method), 61

list\_submissions() (canvasapi.section.Section method), 96

list\_tabs() (canvasapi.course.Course method), 61

list\_tabs() (canvasapi.group.Group method), 79

list\_topic\_entries() (canvasapi.discussion\_topic.DiscussionTopic method), 68

list\_user\_logins() (canvasapi.account.Account method), 36

list\_user\_logins() (canvasapi.user.User method), 103

list\_user\_participants() (canvasapi.Canvas

method), 26

list\_users() (canvasapi.group.Group method), 79

list\_users() (canvasapi.group.GroupCategory method), 82

Login (class in canvasapi.login), 82

## M

mark\_as\_read() (canvasapi.discussion\_topic.DiscussionEntry method), 67

mark\_as\_read() (canvasapi.discussion\_topic.DiscussionTopic method), 69

mark\_as\_unread() (canvasapi.discussion\_topic.DiscussionEntry method), 67

mark\_as\_unread() (canvasapi.discussion\_topic.DiscussionTopic method), 69

mark\_entries\_as\_read() (canvasapi.discussion\_topic.DiscussionTopic method), 69

mark\_entries\_as\_unread() (canvasapi.discussion\_topic.DiscussionTopic method), 69

mark\_read() (canvasapi.submission.Submission method), 97

mark\_submission\_as\_read() (canvasapi.course.Course method), 61

mark\_submission\_as\_read() (canvasapi.section.Section method), 96

mark\_submission\_as\_unread() (canvasapi.course.Course method), 61

mark\_submission\_as\_unread() (canvasapi.section.Section method), 96

mark\_unread() (canvasapi.submission.Submission method), 98

merge\_into() (canvasapi.user.User method), 104

MigrationIssue (class in canvasapi.content\_migration), 47

Migrator (class in canvasapi.content\_migration), 47

Module (class in canvasapi.module), 82

ModuleItem (class in canvasapi.module), 84

## O

Outcome (class in canvasapi.outcome), 84

OutcomeGroup (class in canvasapi.outcome), 85

OutcomeLink (class in canvasapi.outcome), 85

## P

Page (class in canvasapi.page), 88

PageRevision (class in canvasapi.page), 89

parent\_id (canvasapi.external\_tool.ExternalTool attribute), 71

parent\_id (*canvasapi.page.Page* attribute), 88  
 parent\_id (*canvasapi.page.PageRevision* attribute), 89  
 parent\_type (*canvasapi.external\_tool.ExternalTool* attribute), 71  
 parent\_type (*canvasapi.page.Page* attribute), 89  
 parent\_type (*canvasapi.page.PageRevision* attribute), 89  
 post\_entry() (*canvasapi.discussion\_topic.DiscussionTopic* method), 69  
 post\_reply() (*canvasapi.discussion\_topic.DiscussionEntry* method), 67  
 preview\_html() (*canvasapi.course.Course* method), 62  
 preview\_html() (*canvasapi.group.Group* method), 79  
 Progress (*class in canvasapi.progress*), 89

## Q

query() (*canvasapi.progress.Progress* method), 90  
 Quiz (*class in canvasapi.quiz*), 90  
 QuizExtension (*class in canvasapi.quiz*), 92  
 QuizGroup (*class in canvasapi.quiz*), 93  
 QuizQuestion (*class in canvasapi.quiz*), 92  
 QuizSubmission (*class in canvasapi.quiz*), 92

## R

rate() (*canvasapi.discussion\_topic.DiscussionEntry* method), 67  
 reactivate() (*canvasapi.enrollment.Enrollment* method), 70  
 relock() (*canvasapi.module.Module* method), 83  
 remove() (*canvasapi.course.CourseNickname* method), 65  
 remove\_observee() (*canvasapi.user.User* method), 104  
 remove\_self() (*canvasapi.group.GroupMembership* method), 80  
 remove\_user() (*canvasapi.group.Group* method), 79  
 remove\_user() (*canvasapi.group.GroupMembership* method), 80  
 reorder\_pinned\_topics() (*canvasapi.course.Course* method), 62  
 reorder\_pinned\_topics() (*canvasapi.group.Group* method), 79  
 reorder\_question\_group() (*canvasapi.quiz.QuizGroup* method), 93  
 request\_upload\_token() (*canvasapi.upload.Uploader* method), 98  
 reserve\_time\_slot() (*canvasapi.Canvas* method), 26  
 reset() (*canvasapi.course.Course* method), 62

revert\_to\_revision() (*canvasapi.page.Page* method), 89  
 Role (*class in canvasapi.account*), 37  
 Rubric (*class in canvasapi.rubric*), 94

## S

search\_accounts() (*canvasapi.Canvas* method), 26  
 search\_all\_courses() (*canvasapi.Canvas* method), 26  
 search\_recipients() (*canvasapi.Canvas* method), 26  
 Section (*class in canvasapi.section*), 94  
 set\_course\_nickname() (*canvasapi.Canvas* method), 26  
 set\_extensions() (*canvasapi.quiz.Quiz* method), 91  
 set\_quiz\_extensions() (*canvasapi.course.Course* method), 62  
 show\_account\_auth\_settings() (*canvasapi.account.Account* method), 36  
 show\_blueprint\_import() (*canvasapi.blueprint.BlueprintSubscription* method), 42  
 show\_blueprint\_migration() (*canvasapi.blueprint.BlueprintTemplate* method), 43  
 show\_front\_page() (*canvasapi.course.Course* method), 63  
 show\_front\_page() (*canvasapi.group.Group* method), 79  
 show\_latest\_revision() (*canvasapi.page.Page* method), 89  
 show\_observee() (*canvasapi.user.User* method), 104  
 SSOSettings (*class in canvasapi.account*), 37  
 start() (*canvasapi.upload.Uploader* method), 99  
 Submission (*class in canvasapi.submission*), 97  
 submissions\_bulk\_update() (*canvasapi.assignment.Assignment* method), 39  
 submissions\_bulk\_update() (*canvasapi.course.Course* method), 63  
 submissions\_bulk\_update() (*canvasapi.section.Section* method), 96  
 submit() (*canvasapi.assignment.Assignment* method), 39  
 submit\_assignment() (*canvasapi.course.Course* method), 63  
 submit\_assignment() (*canvasapi.section.Section* method), 97  
 subscribe() (*canvasapi.discussion\_topic.DiscussionTopic* method), 69

## T

Tab (*class in canvasapi.tab*), 98

## U

uncomplete() (*canvasapi.module.ModuleItem method*), 84

unlink\_outcome() (*canvasapi.outcome.OutcomeGroup method*), 87

unsubscribe() (*canvasapi.discussion\_topic.DiscussionTopic method*), 70

update() (*canvasapi.account.Account method*), 36

update() (*canvasapi.authentication\_provider.AuthenticationProvider method*), 40

update() (*canvasapi.content\_migration.ContentMigration method*), 46

update() (*canvasapi.content\_migration.MigrationIssue method*), 47

update() (*canvasapi.course.Course method*), 63

update() (*canvasapi.discussion\_topic.DiscussionEntry method*), 67

update() (*canvasapi.discussion\_topic.DiscussionTopic method*), 70

update() (*canvasapi.folder.Folder method*), 73

update() (*canvasapi.group.GroupCategory method*), 82

update() (*canvasapi.group.GroupMembership method*), 80

update() (*canvasapi.outcome.Outcome method*), 84

update() (*canvasapi.outcome.OutcomeGroup method*), 87

update() (*canvasapi.quiz.QuizGroup method*), 93

update() (*canvasapi.tab.Tab method*), 98

update\_account\_auth\_settings() (*canvasapi.account.Account method*), 36

update\_assignment\_overrides() (*canvasapi.course.Course method*), 63

update\_associated\_courses() (*canvasapi.blueprint.BlueprintTemplate method*), 43

update\_color() (*canvasapi.user.User method*), 104

update\_membership() (*canvasapi.group.Group method*), 80

update\_multiple\_preferences() (*canvasapi.communication\_channel.CommunicationChannel method*), 45

update\_preference() (*canvasapi.communication\_channel.CommunicationChannel method*), 45

update\_preferences\_by\_catagory() (*canvasapi.communication\_channel.CommunicationChannel method*), 45

update\_role() (*canvasapi.account.Account method*), 36

update\_score\_and\_comments() (*canvasapi.quiz.QuizSubmission method*), 93

update\_settings() (*canvasapi.course.Course method*), 64

update\_settings() (*canvasapi.user.User method*), 104

update\_submission() (*canvasapi.course.Course method*), 64

update\_submission() (*canvasapi.section.Section method*), 97

update\_tab() (*canvasapi.course.Course method*), 64

upload() (*canvasapi.course.Course method*), 64

upload() (*canvasapi.folder.Folder method*), 73

upload() (*canvasapi.group.Group method*), 80

upload() (*canvasapi.upload.Uploader method*), 99

upload() (*canvasapi.user.User method*), 105

upload\_comment() (*canvasapi.submission.Submission method*), 98

upload\_to\_submission() (*canvasapi.assignment.Assignment method*), 39

Uploader (*class in canvasapi.upload*), 98

User (*class in canvasapi.user*), 99