

UStudy API Documentation

Introduction

UStudy has a robust REST API that allows you to GET and POST data to the server's API endpoints in order to fetch data for your specific use cases relating to UStudy. Requests are stateless and thus the server is able to serve the appropriate JSON response without the need for knowing previous requests.

The API provides a 200 status code when the request and response are error-free, and will provide appropriate error codes otherwise.

Authentication is needed to perform most actions (in this case, a session authentication cookie), but in a few cases, some actions can be performed by anyone.

Signing Up

Signs the user up. Does not require authentication. After calling this end-point, you will likely want to redirect them to the login page to let users log in.

Usage:

```
POST to /api/user/signup
```

Front-end sends:

```
JSON object: { username: string, email: string, password: string }
```

Back-end sends:

```
Status: HTTP 200 (sucess) or HTTP 404 or HTTP 500 error
```

Logging in

Authenticates and logs the user in - both server and client side. This means that the client will have a session token set. After using this end-point, you will likely redirect your user to another page (one which requires authentication presumably).

Usage:

```
POST to /api/user/login
```

Front-end sends:

```
JSON object: { username: string, password: string }
```

Back-end sends:

Authenticated response (set session cookie ID) (status 200 if success, or else HTTP 404 or HTTP 500 error)

Logging out

Unsets the client/user's session token. After calling this end-point, you will likely redirect to your landing page (one which doesn't require authentication)

Usage:

POST to /api/user/logout

Front-end sends:

Nothing

Back-end sends: Status:

HTTP 200 if success, else HTTP 404 or HTTP 500 error

Getting a list of all schools

Returns a list of the schools at the university, and their associated ID.

Usage:

GET to /api/profile/get_all_schools/

Front-end sends:

Nothing

Back-end sends:

List of Schools (id: Int, name: String)

Getting a list of courses within a school

Returns a list of all the courses within a school specified in the parameter.

Usage:

GET to /api/profile/get_courses_in_school/?id={id of school}

Front-end sends:

ID of school in URL

Back-end sends:

List of Courses (id: Int, name: String, school_id: Int)

Accessing your course

Requires authentication.

You can set your course after you have registered and have logged in. You may want to do this before proceeding into other aspects of the API, as this will allow you to retrieve course/module-specific videos that are relevant to you later on (after also setting your user modules).

Usage:

POST to /api/profile/set_my_course/

Front-end sends:

id: Int

Back-end sends:

Success (HTTP 200) or HTTP 404 or HTTP 500 error

In order to then retrieve which course you are taking, you can then do so accordingly.

Usage:

GET to /api/profile/get_my_course/

Front-end sends:

Nothing

Back-end sends:

id: Int, school_id: Int, name: String

Accessing your modules

Requires authentication.

Once you have set your course, you can call this endpoint to automatically populate your enrolled modules.

Usage:

```
POST to /api/profile/set_my_user_modules/
```

Front-end sends:

```
Nothing
```

Back-end sends:

```
Success (HTTP 200) or HTTP 404 or HTTP 500 error
```

Once you have set your specific modules, you can query them as a whole:

Usage:

```
GET to /api/profile/get_all_my_modules/
```

Front-end sends:

```
Nothing
```

Back-end sends:

```
List of Modules (id: Int, name: String)
```

Or you can simply query the modules that you are enrolled in for this academic year (instead of all of them at once)

Usage:

```
GET to /api/profile/get_my_modules_this_year/
```

Front-end sends:

```
Nothing
```

Back-end sends:

```
List of Modules (id: Int, name: String)
```

Listing videos

This section of the API requires authentication.

You can query the API to list the videos of a particular module (such as those that you are enrolled in - but you are also able to peruse videos from other modules as well)

Usage:

```
GET to /api/video/get_module_videos/?id={id of module}
```

Front-end sends:

ID of module in URL

Back-end sends:

List of Videos (Refer to Videos/models.py)

To query the video data of one specific video or all videos, you can query the base endpoint of `/api/video/` for all videos or `/api/video/{ID of video}` for a specific video ID. But you may also choose to use:

Usage:

GET to `/api/video/get_video/?id={id of video}`

Front-end sends:

ID of video in URL

Back-end sends:

id: Int, link: String, user_id: Int, title: String, description: String, has_captions: Bool, when_posted: String (date format)

You can also post a video to UStudy via the API, which also allows you to specify multiple modules that it may come under (as one video may apply to multiple topics across several facets of your course)

Usage:

POST to `/api/video/post_video/`

Front-end sends:

JSON object: { title: String, link: String, description: String, module_ids: List of Int (currently would just be an array of 1 as only 1 module is selectable at a time) }

Back-end sends:

Video ID (if HTTP 200 successful), else HTTP 404 or HTTP 500 error status
