

CSC309 Project Backend Documentation

API Endpoints:

Studios App:

Endpoint: `api/studios/<str:studio_name>/details/`

Methods: GET

Description: View details about `<studio_name>`

Endpoint: `api/studios/list/`

Methods: GET

Query Parameters: lat, lon, amenities, classes, coaches, names

Description: Filter for studios based on name, amenities, classes, and coaches if these parameters are provided. It is possible to filter based on multiple names, amenities, classes, and coaches by separating them with a comma. Filtering by multiple names returns studios (since a studio cannot have more than one name) with ANY of the names queried whereas filtering by multiple amenities, classes, and coaches returns studios with ALL properties queried. View details about filtered studios in a list sorted by distance from (lat,lon). If lat and lon are not provided in the request, then lat and lon are defaulted to (0.0,0.0).

Endpoint: `api/studios/<str:studio_name>/classes/list/`

Methods: GET

Query Parameters: names, dates, coaches, start, end

Description: View the class schedule for `<studio_name>`. Returns a 404 status code if the studio does not exist. Filter the classes in `<studio_name>` based on the name of the class, the date, the start time, and the end time. It is possible to filter based on multiple names, dates, and coaches by separating them with a comma. Lists all classes in studio if no parameters are provided.

Endpoint: `api/studios/classes/enroll/`

Methods: POST

Fields/Payload: `class_id: Int, enroll_future: Bool`

Validation Errors:

- Enrollment Unsuccessful: This class has reached capacity
- Enrollment Unsuccessful: User is already enrolled in this class
- Enrollment Unsuccessful: This datetime of this class has already passed
- User does not have an active subscription

Description: Enroll the user in <class_id>. Returns a 404 status code if the class does not exist. If enroll_future is set to true, this will also automatically enroll user into any class instances that share the same parent_class attribute. Enroll_future is optional and defaults to False

Endpoint: api/studios/classes/unenroll/

Methods: POST

Fields/Payload: class_id: Int, enroll_future: Bool

Validation Errors:

- Unenrollment Unsuccessful: User is not enrolled in this class
- Unenrollment Unsuccessful: This datetime of this class has already passed

Description: Unenroll the user from <class_id>. Returns a 404 status code if the class does not exist. If enroll_future is set to True, this will also automatically unenroll user from any class instances that share the same parent_class attribute. Enroll_future is optional and defaults to False.

Endpoint: api/studios/schedule/

Methods: GET

Query Parameters: history: bool (default=False)

Description: Show the user's schedule of classes they are enrolled in in chronological order. By default, only show classes in the future. If history is set to True, then include past classes.

Endpoint: api/studios/<str: studio_name>/directions

Methods: GET

Description: Redirects to a google maps page with the destination set to the studio location indicated by studio_name. The origin is not specified at the moment.

Accounts App:

Endpoint: api/accounts/subscription/add/

Methods: POST

Fields/Payload: sub_type

Validation Errors:

- Request Not Completed: User does not have payment info
- Request Not Completed: User already has a subscription
- Request Not Completed: Subscription type does not exist

Description: Add a subscription to the user's account of type <sub_type>. Returns a 404 status code if <sub_type> does not exist. <sub_type> options are either 'yearly' or 'monthly'.

Endpoint: api/accounts/subscription/edit/

Methods: PATCH

Fields/Payload: sub_type

Description: Edit the user's subscription to 'monthly' or 'yearly'. Return a 404 status code if <sub_type> is not either of these options or doesn't exist.

Endpoint: api/accounts/subscription/delete/

Methods: POST

Validation Errors:

- User does not have an active subscription to delete

Description: Delete the user's subscription from their account. Also delete the enrollments that the user had created.

Endpoint: api/accounts/payment-info/add/

Methods: POST

Fields/Payload: credit_num, credit_exp_month, credit_exp_year, credit_cvv

Validation Errors:

- Request Unsuccessful: Account has payment info added already

Description: Add payment info to user's account

Endpoint: api/accounts/payment-info/edit/

Methods: PATCH

Fields/Payload: credit_num, credit_exp_month, credit_exp_year, credit_cvv

Validation Errors:

- Request Unsuccessful: Account does not have payment info

Description: Edit user's payment info with provided fields

Endpoint: api/accounts/register/

Methods: POST

Fields/Payload: email, password, repeat_password, first_name, last_name, phone_num, avatar

Description: Create an account. All are required but avatar. Returns new account details and auth token

Endpoint: api/accounts/login/

Methods: POST

Fields/Payload: email, password

Description: Log into an account. Returns an authentication token.

Endpoint: api/accounts/logout/

Methods: POST

Description: Log out of the account if it is logged in. Return authentication missing or invalid authentication depending on issue.

Endpoint: api/accounts/edit/

Methods: PUT

Fields/Payload: email, password, repeat_password, first_name, last_name, phone_num, avatar

Description: Edits all parameters that were sent/not empty. Password only changed if basic password validations match.

Endpoint: `api/accounts/details/`

Methods: GET

Description: Returns the users' email, first name, last_name, and phone number.

Note about editing classes:

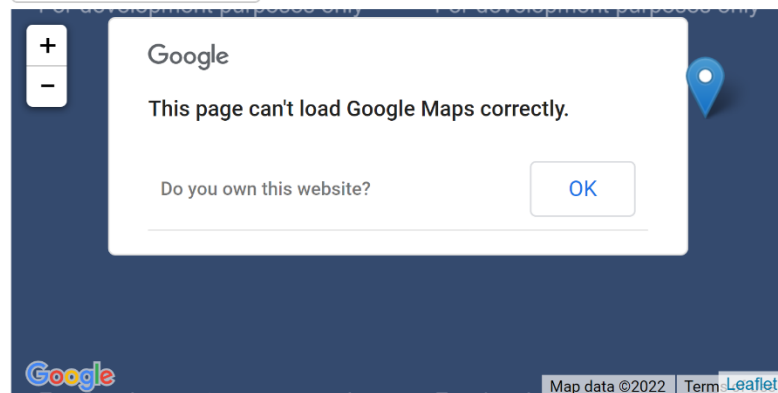
The start and end time of recurring classes can be edited altogether, and individual classes can be edited separately. When editing recurring classes, only the class instances that follow the pre-established pattern (classes that have not been individually edited) will be updated. Classes that have already passed will not be updated when changing start or end time.

Note about creating studios:

To select a location, click on the map, press tab 4 times, press enter, then zoom out until you can see land. The page cannot load Google Maps correctly because we are using the unpaid version of the Google Maps API

Location:

0.009326936519484502,(



Note about “automatic payments”:

Run `python manage.py make_payments` to make payments that are today.

Database Model:

