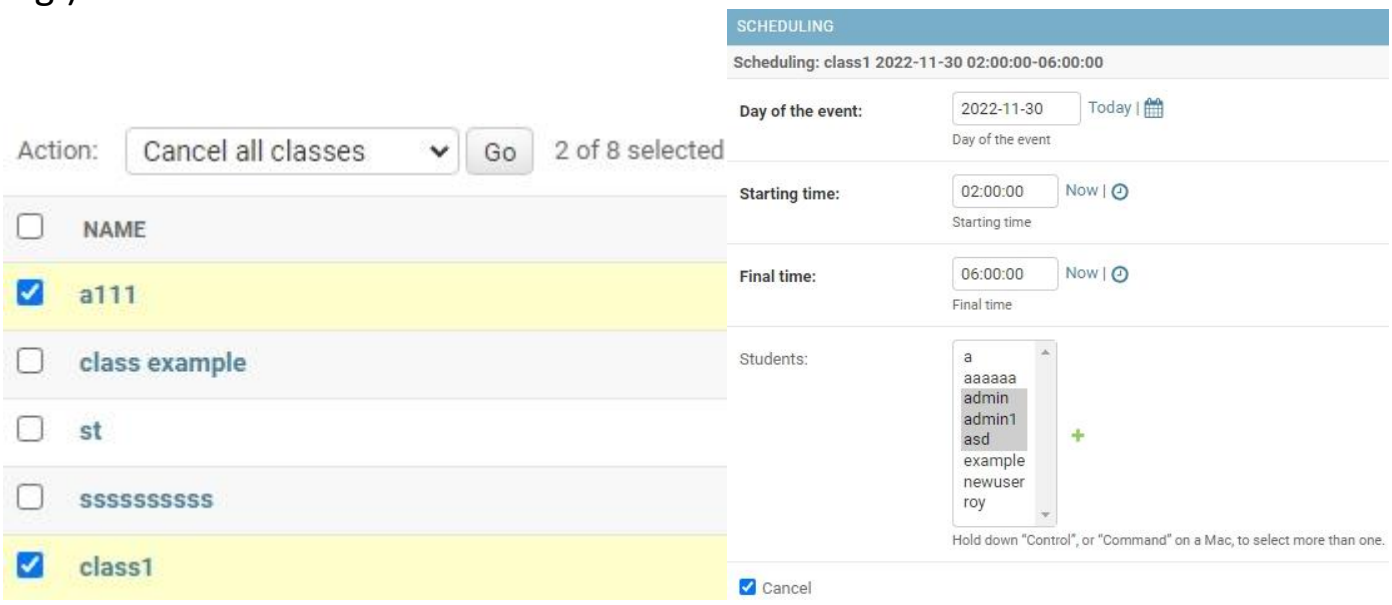
Notes before Startup:

Our Django project is located in tfc folder which is our project name. Since all shell scripts are located in root folder, we made shell scripts to change directory to root directory after a script is finished. If you want to create a super user, you need to change directory to tfc folder and run manage.py in the folder. To run a run.sh script, you should come back to the root folder.

How to cancel a specific class in the admin panel:

Website admin can cancel a specific class; either one instance or all the future occurrences of the class. To cancel all the future occurrences of the class, admin must use Django admin action. To cancel one instance of the class, you can check an occurrence's cancel to be true.

e.g.)



How to check and make the payments that are due today:

1. Django management command: manage.py renew
2. Shell script: run renew.sh in the root folder

Note: They also cancel class bookings of users who don't renew subscriptions

Endpoints:

All endpoints come after http://127.0.0.1:8000

a) Subscriptions:

1) User Subscription plan create:

Endpoint: /api/subscriptions/
Methods: POST
Headers: token {{token}}
Fields/payload: user,username,curr_plan,renew,last_paid,next_pay,cardnumber,cvv,expiry
Notes: Allows user to create a user subscription only requires card details, plan, and choice if renewal. Only logged in User can acesss, throws unauthorized error if not logged in. Payload "curr_plan" should be an id of plan.

2) User Subscription plan view edit and delete:

Endpoint: /api/subscriptions/{{username}}/
Methods: GET,PUT,DELETE
Headers: token {{token}}
Fields/payload: user,username,curr_plan,renew,last_paid,next_pay,cardnumber,cvv,expiry
Notes: Allows user to edit user subscription throws unauthorized error if not logged in. If change in plan, then new payment is made, also updates card info. Payload "curr_plan" should be an id of plan.

3) Payments History:

Endpoint: /payments /{{username}}/
Methods: GET
Headers: token {{token}}
Fields/payload: username
Notes: Allows user to view all previous payments of given username throws unauthorized error if not logged in.

b) Accounts:

1. Create Accounts:

Endpoint: /api/accounts/
Methods: POST
Fields/payload: username,password,password2,email_address,first_name,last_name,phone_number, avatar
Notes: Allows user to create account

2. Accounts profile view and edit:

Endpoint: /api/accounts/{{username}}/
Methods: GET,PUT
Fields/payload: user,username,curr_plan,renew,last_paid,next_pay,cardnumber,cvv,expiry
Notes: Allows user to go to profile, view and edit account details

3. Login:

Endpoint: /api/accounts/login/
Methods: POST
Fields/payload: username, password
Notes: Allows user to login

4. Logout:

Endpoint: /api/accounts/logout/
Methods: POST
Headers: token {{token}}
Fields/payload: None
Notes: Allows user to logout.

c) Studios:

1. All studio list:

Endpoint: /studios/
Methods: GET
Fields/payload: None
Notes: You can get a list of all studios

2. Studio detail:

Endpoint: /studios/{{studio_id}}/
Methods: GET
Fields/payload: None
Notes: This page contains the general information of that studio, along with its address, location, and a link to get the directions. Also, User can see the class schedule of a specific studio on its page. Classes appear in the order of their start time (from now), and the class information must be shown. Past or cancelled classes should not be listed.

3. List of studios based on geolocation:

Endpoint: /studios/{{latitude}}/{{longitude}}/
Methods: GET
Fields/payload: None
Notes: This page gives a list studios based on geographical locations. Studios are listed starting from the closest one.

d) Classes:

1. Class Enroll

Endpoint: /api/accounts/{{username}}/enrol/
Methods: POST
Fields/payload: class_id, instance_id
Notes: Allows user to enroll in class only if user is logged in and has active subscription. If instance_id is provided, user can enroll a single instance of a class (doesn't matter what class_id is).

2. Class Drop

Endpoint: /api/accounts/{{username}}/drop/
Methods: POST
Fields/payload: class_id, instance_id
Notes: Allows user to drop in class only if user is logged in. If instance_id is provided, user can drop a single instance of a class (doesn't matter what class_id is).

3. User Class Schedules

Endpoint: /api/accounts/{{username}}/schedule/
Methods: GET
Fields/payload: None
Notes: Allows user to see enrolled classes schedules

4. User Class History

Endpoint: /api/accounts/{{username}}/history/
Methods: GET
Fields/payload: None
Notes: Allows user to see classes history

5. Class schedule of a specific studio

Endpoint: /studios/{{studio_id}}/
Methods: GET
Fields/payload: None
Notes: User can see the class schedule of a specific studio on its page. Classes appear in the order of their start time (from now), and the class information must be shown. Past or cancelled classes are not listed.

e) Search and Filtering

1. Studio's class schedule filter by class name:

Endpoint: /studios/{{studio_id}}/class/{{class_name}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using class name.

2. Studio's class schedule filter by coach name:

Endpoint: /studios/{{studio_id}}/coach/{{coach_name}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using coach name.

3. Studio's class schedule filter by time range:

Endpoint: /studios/{{studio_id}}/time/{{hour}}-{{min}}-{{sec}}/{{hour2}}-{{min2}}-{{sec2}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using time range.

4. Studio's class schedule filter by date range:

Endpoint: /studios/{{studio_id}}/date/{{year}}-{{month}}-{{day}}/{{year2}}-{{month2}}-{{day2}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using date range.

5. Studio's class schedule filter by start date:

Endpoint: /studios/{{studio_id}}/date/{{year}}-{{month}}-{{day}}/start/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using start date. (start after provided date)

6. Studio's class schedule filter by end date:

Endpoint: /studios/{{studio_id}}/date/{{year}}-{{month}}-{{day}}/end
Fields/payload: None
Notes: As a user, I can search/filter a studio's class schedule using end date. (end before provided date)

7. Search through the listed studio using studio name:

Endpoint: /studios/{{latitute}}/{{longitude}}/?search={{studio_name}}
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter through the listed studios using studio name.

8. Search through the listed studio using class name:

Endpoint: /studios/{{latitute}}/{{longitude}}/class/{{class_name}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter through the listed studios using class name.

9. Search through the listed studio using coach name:

Endpoint: /studios/{{latitute}}/{{longitude}}/coach/{{coach_name}}/
Methods: GET
Fields/payload: None
Notes: As a user, I can search/filter through the listed studios using coach name.

10. Search through the listed studio using amenity:

Endpoint: /studios/{{latitute}}/{{longitude}}/amenities/{{amenities_type}}/

Methods: GET

Fields/payload: None

Notes: As a user, I can search/filter through the listed studios using amenity.