Model Information

Model: PaymentInfo

Description:

Data oriented model holding credit card information for one user.

Fields:

cc_number: string representing user credit card number
cc_expiry: string representing user credit card expiry date
cc_code: string representing user credit card expiry date

Model: Avatar

Description:

Data oriented model holding image data for user avatar

Fields:

image: image representing user avatar, uses django PILLOW python imaging library to handle user image data

Model: GUser

Description:

Represents a non-admin user of TFC. Maintains a One-to-One relation with native Django user object to make user of built in authorization and authentication features whilst holding additional fields for essential user data.

Fields;

user: maintains one to one relation with native Django User object
avatar: maintains one to one relation with Avatar object holding user avatar image
Phone_number: string representing user phone number
classes: many to many relation with Class objects, representing user's classes
class_occurrences: many to many relation with ClassOccurrence objects, represents users scheduled and previously scheduled classes
payment_info: one to one relation with PaymentInfo object storing users credit card info
subscription: many to one relation with Subscription object representing users current subscription
address: string representing users address, validated using Geopy geolocation library

Model: Subscription

Description:

Represents a gym subscription to the gyms of TFC. Contains the type of subscription: monthly or yearly, and the cost of the subscription, in dollars. GUsers have a many-to-one relationship to a Subscription object.

Fields:

type: the time frame of the subscription: monthly or yearly
cost: the amount of money that the subscription costs per time frame

Model: Transaction

Description:

Represents a payment that TFC was / will be charged to the user and contains information about when the amount was/will be charged.

Fields:

user: the user that the amount is charged to. Transaction has a many-to-one relationship to the User object
amount: the amount that was / will be charged
timestamp: the time and date that the payment was / will be charged

Model: Studio

Description:

Represents a studio (i.e., individual location) of the TFC gym corporation. Contains all relevant information regarding its geographical location and administrative details

Fields:

name: the name of the studio
address: The street address of the studio (e.g., 100 Queen St. W)
latitude: The geographical latitude for the studio's location
longitude: The geographical longitude for the studio's location
postal_code: The studio's postal code
phone_number: The phone number for the studio's office/front desk

Model: Image

Description:

Represents an image of a studio. Exists as a separate model in order to maintain ForeignKey relationship to Studio (allows for multiple images that all correspond to one studio)

Fields:

studio: the studio that these images are for
images: the images that showcase the studio

Model: Amenity

Description:

Represents an amenity within a studio. Amenities will typically be extra perks of facilities that can be different across different studios.

Fields:

studio: The studio that this amenity belongs to
type: describes what the amenity is, in other words, its name (e.g., Massage Room)
quantity: The amount of this type amenity that exists in the studio

Model: Class

Description:

Represents a class that is held at a specific studio. Classes are sessions or activities held by a specific coach and can either be a one-time instance, or can be recurring (occurring in some pattern, typically at the same time - e.g., Every Monday at 7-8pm, daily at 5-6pm for 1 week)

Fields:

studio: the name of the studio that this class is held in
name: the name of the class
description: a short summary of what the class is about or what will be going on
coach: the name of the coach who will be leading the class
capacity: the maximum amount of people that can be enrolled in each class (positive integer)
start_datetime: the date and time of the beginning of the first class (datetime value)
end_datetime: the date and time of the end of the last class (datetime value)
duration: the length of the class in hours (positive hour value)
is_recurring: boolean denoting whether the class is a recurring class or only one-time
recurrences: an RRULE (recurrence rule) defining the the recurrence pattern for this class (e.g., every monday from start_datetime to end_datetime) (makes use of django-recurrence library)

Model: ClassOccurrence

Description:

Represents a specific instance of a class (e.g., Yoga on Monday, November 21, 2022, at 7-8 pm).

Fields:

parent_class: the type of Class, or more specifically, the Class object that this instance corresponds to (e.g., if this is an instance of a Yoga class, parent class is Yoga) - foreign key to Class
start_datetime: the date and time of the start of this class (datetime value)
end_datetime: the date and time of the end of this class (datetime value)
num_attending: the number of people attending this instance of the class (must be less than or equal to the capacity field in the parent_class) (positive integer)

Model: Keyword

Description:

A specific word that can be used to describe what goes on in a specific class. Keywords have a foreign key relationship to classes, so a class can have multiple keywords that describe it.

Fields:

ex_class: the class that this keyword corresponds to (foreign key to Class)
word: the keyword itself, which describes the class

☰ Documentation Settings ▼

309 Project API

CSC309 Project Backend API Documentation

This documentation contains the endpoints, along with descriptions and examples of payloads, for all the functionality defined in the Project's User Stories.

POST Register New User (Create)

```
http://127.0.0.1:8000/accounts/register/
```

Create a new user in the system. View for endpoint creates a GUser object which is our group's extension of the default User model provided by Django.

```
Request Body: (Form Data)
    username: string
    first_name: string
    last_name: string
    email: string (valid email address - example@example.com)
    password: string
    avatar: image_url
    phone_number: string | Format: +1XXXYYYZZZZ
    address: string | Format: XXX Street Name St., City, Province Code
```

View More

BODY formdata

username

patientzero2

first_name

Patient

last_name

Zero

email

Sickness123$

phone_number

+14167236514

address

100 Queen St. W

Example Request Register New User (Create)

```
curl --location --request POST 'http://127.0.0.1:8000/accounts/register/' \
--form 'username="patientzero2"' \
--form 'first_name="Patient"' \
--form 'last_name="Zero"' \
--form 'email="patientzero@uhn.ca"' \
--form 'password="Sickness123$"' \
--form 'phone_number="+14167236514"' \
--form 'address="100 Queen St. W"'
```

POST User Login (Get Auth Token)

http://127.0.0.1:8000/accounts/api/token/

Endpoint for user to send a login request to. They must provide their username and password into the request body as form data, which will be authenticated by the server. Upon success, user will receive an access token and a refresh token. Upon failure, they will receive an error message and must try logging in again. This endpoint uses DRF's simple-jwt library to handle all token operations.

```
Request Body: (Form data)
    username: string
    password: string
Authorization: None
Response Body: Payload
Successful response:
{
    "refresh": "refresh_token_string",
    "access": "access_token_string"
```

View More

☰ Documentation Settings ▼

username

patientzero1

password

Sickness123$

Example Request User Login (Get Auth Token)

```
curl --location --request POST 'http://127.0.0.1:8000/accounts/api/token/' \
--form 'username="patientzero1"' \
--form 'password="Sickness123$"'
```

POST Token Refresh

http://127.0.0.1:8000/accounts/api/token/refresh/?

Endpoint that allows user to make a request to get a new access_token if their old one expires (handled by the simple-jwt library).

```
Request body: (Form Data)
    refresh "refresh_token_string"
Authorization: None
Response body: (Payload)
Successful Response (200 OK)
{
    "access": "access_token_string"
}
Failure Response (401 Unauthorized)
```

View More

PARAMS

BODY formdata

refresh

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTY2ODk2ODQxMiwiaWF0IjoxNjY4ODgy

☰ Documentation Settings ▾

Example Request Token Refresh

```
curl --location --request POST 'http://127.0.0.1:8000/accounts/api/token/refresh/' \
--form 'refresh="eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MT
```

GET View User Profile Details

http://127.0.0.1:8000/accounts/profile/

View user profile information; first and last name, email, avatar, and phone number.

```
Request Body: (None)
Authorization: Bearer Token
Response Body: (Payload)
Successful Response (200 OK):
Success response:
{
    "first_name": "user_first_name",
    "last_name": "user_last_name",
    "email": "user@example.com",
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

Example Request View User Profile Details

☰ Documentation Settings ▼

PATCH Edit User Profile Details

http://127.0.0.1:8000/accounts/profile/

```
Request Body: (Form Data)
    first_name: string
    last_name: string
    email: string (valid email address - example@example.com)
    avatar: image_file
    phone_number: string | Format: +1XXXYYYZZZZ
Authorization: Bearer Token
Response Body: Payload
Success response:
```

View More

Edit any number of user profile details; first and last name, email, avatar, and phone number.

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

BODY formdata

first_name

Fat

last_name

Sunny

email

django@unc.al

phone_number

+14165514567

☰ Documentation Settings ▼

Example Request Edit User Profile Details

```
curl --location --request PATCH 'http://127.0.0.1:8000/accounts/profile/' \
--form 'first_name="Fat"' \
--form 'last_name="Sunny"' \
--form 'email="django@unc.al"' \
--form 'phone_number="+14165514567"' \
--form 'avatar=@"/Users/lazarus/Downloads/wallpaperflare.com_wallpaper.jpg"'
```

GET View User Classes Schedule and HIstory

http://127.0.0.1:8000/accounts/profile/classes/

Lists logged in users class schedule and history in chronological order.

```
Request body: None
Authorization: Bearer Token
Response Body (Payload):
Sample Successful Response (200 OK):
{
    "count": 11,
    "next": "http://127.0.0.1:8000/studios/3/classes/schedule/?page=2",
    "previous": null,
    "results": [
```

View More

AUTHORIZATION Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

BODY formdata

☰ Documentation Settings ▼

Example Request View User Classes Schedule and HIstory

```
curl --location --request GET 'http://127.0.0.1:8000/accounts/profile/classes/'
```

GET View User Payment information

http://127.0.0.1:8000/accounts/profile/payment/

View the current payment details for logged in User.

```
Request Body: (None)
Authorization: Bearer Token
Response Body: (Payload)
Successful Response (200 OK):
Success response:
{
    "cc_number" : "4716495763552956"
    "cc_expiry" : "2002-03-05"
    "cc_code" :  "123"
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

Example Request View User Payment information

```
curl --location --request GET 'http://127.0.0.1:8000/accounts/profile/payment/'
```

☰ Documentation Settings ▼

```
http://127.0.0.1:8000/accounts/profile/payment/
```

Edit some or all of a users payment information, including credit card number, expiry date and security code.

```
Request Body: (Form Data)
    cc_number: String | valid credit card number
    cc_expiry: String | format: YYYY-MM-DD
    cc_code: String | valid three number security code
Authorization: Bearer Token
Response Body: Payload
Successful Response (200 OK):
Success response:
{
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

BODY formdata

cc_number

4716495763552956

cc_expiry

2025-05-09

cc_code

123

Example Request Edit User Payment Information

```
curl --location --request PATCH 'http://127.0.0.1:8000/accounts/profile/payment/' \
--form 'cc_number="4716495763552956"' \
```

☰ Documentation Settings ▾

GET View Studio Details (by ID)

```
http://127.0.0.1:8000/studios/:id/details/
```

Displays details for a particular studio, including name, location information, phone number, and a link to directions from user's address to studio in Google Maps.

```
Request Body: (None)
Authorization: Bearer Token
Response Body: (Payload)
Successful Response (200 OK):
{
    "name": "studio_name",
    "address": "studio_address",
    "latitude": "studio_lat",
    "longitude": "studio_long",
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PATH VARIABLES

id

3

Example Request View Studio Details (by ID)

```
curl --location --request GET 'http://127.0.0.1:8000/studios/3/details/'
```

☰ Documentation Settings ▼

GET List All Studios

http://127.0.0.1:8000/studios/all/

For a logged in user only, display all of TFC's studios in order of geographical proximity to user (closest to farthest).

```
Request Body: none
Authorization: Bearer Token
Response Body: Payload
Successful response (200 OK):
[
    {
        "name": "Studio 1",
        "address": "Studio 1 Address",
        "latitude": "XX.XXXXXX",
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

Example Request List All Studios

```
curl --location --request GET 'http://127.0.0.1:8000/studios/all/'
```

GET View Class Details (by ID)

http://127.0.0.1:8000/studios/:studio_id/classes/:class_id/details/

☰ Documentation Settings ▼

```
Request body: None
Authorization: Bearer Token
Response Body: (Payload)
Successful Response (200 OK):
{
    "id": 3,
    "name": "Spin Class",
    "description": "Get your cardio in with some cycling!",
    "coach": "1",
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PATH VARIABLES

studio_id

3

class_id

3

Example Request View Class Details (by ID)

```
curl --location --request GET 'http://127.0.0.1:8000/studios/3/classes/3/details/'
```

GET List Classes for Studio

```
http://127.0.0.1:8000/studios/:studio_id/classes/all/
```

Lists the details for all classes held by the studio corresponding to the value of the "studio_id" path variable.

```
Response Body (payload)
Sample successful response (200 OK):
{
    "count": 2,
    "next": null,
    "previous": null,
    "results": [
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PATH VARIABLES

studio_id

3

Example Request List Classes for Studio

```
curl --location --request GET 'http://127.0.0.1:8000/studios/3/classes/all/'
```

GET View Class Schedule

```
http://127.0.0.1:8000/studios/:id/classes/schedule/
```

Retrieves a list of all instances of each class for the studio corresponding to the value of the path variable "studio_id". Shows key details for instances of all classes, and orders them by order in which they occur (ascending order of datetime).

```
Request body: None
Authorization: Bearer Token
Response Body (Payload):
```

☰ Documentation Settings ▾

```
Sample Successful Response (200 OK):
{
    "count": 11,
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PATH VARIABLES

id

99

Example Request View Class Schedule

```
curl --location --request GET 'http://127.0.0.1:8000/studios/99/classes/schedule/'
```

POST Drop/Enrol in Class as User

http://127.0.0.1:8000/studios/:studio_id/classes/:class_id/enrollment/

If "action" is "enrol", enrol in a specific class (corresponds to the one in studio with id "studio_id" and with class id "class_id") if the user is not already enrolled in it. Adds this class to the set of classes the logged in user is attending. If action is "drop", drop this class for the logged in user if they are enrolled in it. If "all" is set to true, no dates need to be provided; user will be either enrolled to all future dates, or dropped from all future dates. If "all" is "false", dates must be provided in the format "2022-11-18T16:00:00Z" into the list "dates".

```
Response body: JSON
Samples:
{
```

☰ Documentation Settings ▾

```
}
{
    "action": "drop",
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PATH VARIABLES

studio_id

3

class_id

3

BODY raw

```
{
    "action": "enrol",
    "all": "false",
    "dates": ["2022-11-18T16:00:00Z", "2022-11-25T16:00:00Z"]
}
```

Example Request Drop/Enrol in Class as User

```
curl --location --request POST 'http://127.0.0.1:8000/studios/3/classes/3/enrollment/' \
--data-raw '{
    "action": "enrol",
    "all": "false",
    "dates": ["2022-11-18T16:00:00Z", "2022-11-25T16:00:00Z"]
}'
```

☰ Documentation Settings ▾

```
http://127.0.0.1:8000/studios/:studio_id/classes/filter
```

Filters all of the classes for the studio given by the path variable "studio_id", by date, name of the class, or coach name (only one criterion at a time, cannot mix multiple search terms).

```
Request Body: None
Query Parameter Options (one at a time except for datetime, start and end must be used together
    start_date_time:yyyy-mm-ddThh:mm:ssZ
    end_date_time:yyyy-mm-ddThh:mm:ssZ
    class_name=String
    coach=String
Authorization: Bearer Token
Sample Response Body:
Success (200 OK):
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PARAMS

start_date_time

2022-11-16T18:00:00Z

end_date_time

2022-11-30T19:00:00Z

class_name

Yoga

coach

jack

PATH VARIABLES

studio_id

3

☰ Documentation Settings ▾

Example Request Filter Classes By Single Search Criterion

```
curl --location --request GET 'http://127.0.0.1:8000/studios/3/classes/filter'
```

GET Filter Studios by Single Search Criterion

http://127.0.0.1:8000/studios/filter?coach=Billy Bob

Filters all TFC studios in the system by name of class, coach name, amenity type, or studio name (only filters by one criterion at a time, cannot mix multiple). If a filter is performed for a category that is not listed above, it is treated as a bad search and all studios will be returned.

```
Request Body: None
Query Parameters (one at a time):
    class_name=String
    amenity=String
    coach=String
    studion_name=String
Authorization: Bearer Token
Response Body: Payload
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

PARAMS

class_name

Yoga

coach

Billy Bob

☰ Documentation Settings ▾

massages
studio_name

Bay and College Studio

fake

data

Example Request Filter Studios by Single Search Criterion

```
curl --location --request GET 'http://127.0.0.1:8000/studios/filter?coach=Billy Bob'
```

GET View All Subscriptions

http://127.0.0.1:8000/subscriptions/options/?page=1

Display all of TFC's subscription options in a paginated format: each page contains 10 subscriptions. If the page is not specified, the request defaults to displaying the first page.

```
Request Body: None
Query Parameters:
    page=Integer
Authorization: Bearer Token
Sample Response Body:
Success (200 OK):
{
    "count": 2,
    "next": null,
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

☰ Documentation Settings ▼

page

1

Example Request View All Subscriptions

```
curl --location --request GET 'http://127.0.0.1:8000/subscriptions/options/?page=1'
```

POST Create/Update a Subscription as User

http://127.0.0.1:8000/subscriptions/subscribe/

Create/Update the currently logged-in user's subscription based on what is sent in the request body. Returns a JSON representation of a Transaction.

Fails when the request body is invalid, the user is not authenticated, or the user attempts to subscribe to their current plan.

In the case of update, all future payments are cancelled and replaced.

When the user selects a monthly subscription, 12 Transaction objects are made to represent the next 12 payments for the year. Each transaction is spaced a month apart, starting from the current date.

When the user selects a yearly subscription, 2 Transaction objects are made to represent the payment for this year and the next. The transactions are spaced a year apart, starting from the current date.

```
Request Body:
    type=String
Query Parameter Options: None
Authorization: Bearer Token
Success (200 OK):
{
    "amount": "19.99",
    "timestamp": "2022-11-18T23:23:58.444159Z"
}
```

View More

AUTHORIZATION

Bearer Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

BODY formdata

type

monthly

type

yearly

Example Request Create/Update a Subscription as User

```
curl --location --request POST 'http://127.0.0.1:8000/subscriptions/subscribe/' \
--form 'type="monthly"'
```

POST Cancel the User Subscription

http://127.0.0.1:8000/subscriptions/cancel/

Cancels the currently logged-in user's subscription. Deletes all future payments and invalidates all class bookings after the current billing period.

Fails if the user doesn't have a subscription or if the user is not authenticated.

```
Request Body: None
Query Parameter Options: None
Authorization: Bearer Token
Sample Response Body:
Success (200 OK):
{
    "your subscription": null
}
Failure (400 Bad Request):
```

View More

☰ Documentation Settings ▾

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

Example Request Cancel the User Subscription

```
curl --location --request POST 'http://127.0.0.1:8000/subscriptions/cancel/'
```

GET View Payment History

http://127.0.0.1:8000/accounts/profile/payment-history

Displays the logged-in user's payment history and the upcoming payment, if it exists.

```
Request Body: None
Query Parameter Options: None
Authorization: Bearer Token
Sample Request Body:
Success (200 OK):
{
    "payment_history": [
        {
            "amount": 19.99,
```

View More

AUTHORIZATION

Bearer Token

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjY4NzkzOTY3LCJpYXQiOjE2Njg3OTM2NjcsImp0aSI6IjI5ZDM4MWJmNGNiOTRjYTc4ZjUyYWIzY2QxNzFkZWViIiwidXNlcl9pZCI6M30.lH38SOxeYHpfM6P2sMy6QRFMhYPEOP457O52hJxSbgU

☰

Documentation Settings ▼