

Backend Project Documentation
Aria Paydari, Eric Chalmers, Noah Subedar

Instructions

1. Clone the repo
2. Go into the repo directory, go into PB folder
3. Run the command `bash startup.sh`
4. Please make a super user in order to access the admin panel
5. Run the command `bash run.sh`

Continued Below...

Models

Member Model

Description

The member model represents each member of the club. This model effectively replaces the default Django user model.

Fields

Field Name	Description	Example
user	A one-to-one relationship with the default Django user model. The username, password, first name, and last name are stored in this field during registration.	
timestamp	The date when the member was created. Automatically generated upon registration.	
phone_number	The member's phone number. Must be a 10 digit positive integer.	4162501145
avatar	A string representing a link to the user's avatar. It must be type .png or .jpg	"sample.com/image.png"
classes	A many-to-many field containing the classes the member has signed up for.	
credit_card	The user's credit card. Must be in a string in the format of a 16 digit integer (spaces accepted)	"3848 3940 3842 2941"
subscription	The user's subscription to TFC. This is a one-to-one relation with the subscription model.	
payment_history	<p>A string representing the payment history. This is split into two sections, previous payments, and future payments. Previous payments are in the format:</p> <ul style="list-style-type: none">• "yyyy-mm-dd HH:MM \$amount #"• Ex:<ul style="list-style-type: none">○ 2022-09-19 17:44 \$15.23 3856294716583591 <p>Future payments are in the format:</p> <ul style="list-style-type: none">- "yyyy-mm-dd HH:MM \$amount" (all separated by commas)	
class_history	A string representing the history of the classes which the member has enrolled in.	

Class Model

Description

The Class model represents a class that takes place in a studio.

Fields

Field Name	Description	Example
name	the name of the class. Must be unique for each studio	
description	The description of the class.	
keywords	A list of keywords for the class, which describes what the exercises are used for.	“Upper body, Biceps”
capacity	The remaining capacity of the class. Cannot be under 0.	
coach	the name of the class’s coach.	
times	The times in which the class occurs each week. The saved format is: <ul style="list-style-type: none">- Day Start time End time Start date End date- "day of week HH:MM HH:MM MMDDYY MMDDYY"	“Friday 14:15 15:20 111822 121622”

Studio Model

Description

The studio model represents one studio of TFC, which can have several classes.

Fields

Field Name	Description	Example
name	The name of the studio.	“TFC Dundas”
address	The studio’s address.	“308 Negra Arroyo Lane”
location	The latitude and longitude of the studio. Must be in the format “latitude,longitude”	“43.6596,-79.3969”
postal_code	The postal code address of the studio, used for location and distance filtering. Must be unique, and in the format “letter-number-letter-number-letter-number”	“L7H4F1”
phone_number	The phone number of the studio. Must be a 10 digit positive integer0	8672471038
classes	A list of all of the classes offered in a particular studio location.	
images	A string representing a list of images of the studio. Each image must be a url to a .png or .jpg image and each image must be separated by commas	“sample.com/image.png,sample.com/image2.jpg”
amenities	The list of amenities provided in the studio. Must be in the format: “type:quantity,type:quantity, ...”	“treadmill:10,stretching area:50”

Subscription Model

Description

The Subscription model represents a member subscription to TFC, which enables the member to access all studios.

Fields

Field Name	Description	Example
amount	The price of the subscription, which is the amount that the member is charged at each interval (no dollar sign stored)	10.50
plan	The subscription plan, which is the frequency from which the member is charged. Must be one of “Weekly”, “Biweekly”, “Monthly” or “Yearly”.	

Endpoints

NOTE: None of these endpoints that require authentication can be accessed by an admin user other than access amenities.

Endpoint: /accounts/register/

Methods: POST

Authorization Required: No

Fields/payload: username, password, first_name, last_name, email, phone_number, avatar, credit_card

Return value: “Registered” (on success)

Description: Register for a membership at TFC by inputting user information. After successful registration, you can now log in.

Payload restrictions:

- All fields cannot be null
- Email must be in valid email format
- Phone number must be a 10 digit positive integer
- Credit card must be a string of a 16 digit positive number (spaces are ignored)

Endpoint: /accounts/login/

Methods: POST

Authorization Required: No

Fields/payload: username, password

Return value: token (e.x ecaa09b3a147a1630ec465ea78739a973f2a455c)

Description: Login with a previously registered account. A token will be returned, which can be used for authorization of API requests which require authentication

Payload restrictions:

- Username and password must match a previously registered member

Notes:

- Once you receive the token go to the collection menu and click the variables tab, put the token in the “current value” field.

Endpoint: /accounts/profile/

Methods: GET, POST

Authorization Required: Yes

Fields/payload: None (for GET)

first_name, last_name, email, phone_number, avatar (for POST)

Return value: {“first_name”: first_name, “last_name”: last_name, “email”: email, “phone_number”: phone_number, “avatar”: avatar} (on success)

Description: View member profile details (GET)

Edit profile details(POST).

Any fields left empty will retain their previous values. Returns the current (changed or no) member profile details.

Payload restrictions(POST):

- Null fields will result in that member field retaining its previous value
- Email must be in valid email format
- Phone number must be a 10 digit positive integer

Endpoint: /accounts/edit_payment/

Methods: GET, POST

Authorization Required: Yes

Fields/payload: None (for GET)

credit_card (for POST)

Return value: “successfully updated credit card” (on success)

Description: View your current credit card number (GET)

Change credit card number(POST).

Returns the member’s credit card number.

Payload restrictions(POST):

- Credit card must be a string of a 16 digit positive number (spaces are ignored)

Endpoint: /accounts/view_payment/

Methods: GET

Authorization Required: Yes

Fields/payload: None (for GET)

Return value: {“previous_payments”: payment history, “Future installments”: future installments} (on success)

Description: View the member’s payment information. The payment history shows every instance of previous payments, in the format

The future installments are shown in the format

With the interval between the installments depending on the subscription plan.

Payload restrictions: None

Endpoint: /places/nearby_studios/

Methods: POST

Authorization Required: Yes

Fields/payload: latitude, longitude

Return value: [{"name": studio.name,
"address": studio.address,
"location": studio.location,
"postal_code": studio.postal_code,
"phone_number": studio.phone_number,
"images": studio.images,
"amenities": studio.amenities,
"distance": distance},
...
] (on success)

Description: Takes in a location based on latitude and longitude and returns a sorted list of the closest studios. The studios will have their information(name,address, location, postal_code, phone_number, images, amenities) as well as the distance from the location.

Payload restrictions: - latitude and longitude must be of type int or float

- Latitude must be in the range [-90, 90] and longitude must be in the range [-180, 180]

Endpoint: /places/studio_schedule/

Methods: POST

Authorization Required: No

Fields/payload: studio_name

Return value: {"Sunday": [(schedule)],
"Monday": [(schedule)],
"Tuesday": [(schedule)],
"Wednesday": [(schedule)],
"Thursday": [(schedule)],
"Friday": [(schedule)],
"Saturday": [(schedule)]

}(on success)

Description: Takes in the name of a studio in the club and returns its weekly schedule, with each day of the week listing the classes that take place (sorted by end time) alongside its details (name, start time, end time, description, keywords, capacity, instructor name). An example would be:

“Tuesday”: [
[“Aerobics”, “12:30”, “14:15”, “A class for perfecting your aerial balance.”, “balance, upper body strength”, 20, “James McGill”],
[“Yoga”, “14:45”, “16:00”, “A spiritual and historic Indian exercise.”, “relaxation, stretching”, 9, “Saul Goodman”],
]

Payload restrictions: - Studio name must match one of the existing studios

Endpoint: /places/studios/<studio_name>/

Methods: GET

Authorization Required: No

Fields/payload: None

Return value: {"name": studio.name,
"address": studio.address,
"location": studio.location,
"postal_code": studio.postal_code,
"phone_number": studio.phone_number,
"images": studio.images,
"amenities": studio.amenities,
"directions": directions}
(on success)

Description: Returns the <studio_name> studio’s details (name, address, location, postal code, phone_number, image, amenities) alongside “directions” which will be a link to directions to the studio in Google maps. Depending on your browser, the link could automatically fill out the start point of the directions with your location, or else you could click “your location”.

Payload restrictions: None

Endpoint: /places/my_classes/

Methods: GET, POST

Authorization Required: Yes

Fields/payload: None (GET)

Name, request (POST)

Return value: GET: a schedule of all of the classes in which the member is enrolled in, with each being sorted by the day of the week

POST:

“Enrolled” if successfully enrolled in the class

“Already Enrolled” if member already in class

“Registration Date Passed” or “Class is full”

“Not enrolled” if attempted to drop class which user isn’t already enrolled in

“Dropped” if the class is successfully dropped

Description: This endpoint pertains to the member’s classes. The GET request returns a weekly schedule of all of the classes in which the member is enrolled in with their details, e.x

```
{ "Sunday": [ ],  
  "Monday": [ "Aerobics", "12:30", "14:15", "A class for perfecting your aerial balance.", "balance, upper body strength", 20, "James McGill"],  
  "Tuesday": [ ],  
  "Wednesday": [ ],  
  "Thursday": [ ],  
  "Friday": [ "Yoga", "14:45", "16:00", "A spiritual and historic Indian exercise.", "relaxation, stretching", 9, "Saul Goodman"],  
  "Saturday": [ (schedule) ]  
}
```

The POST request will indicate whether the class has been enrolled/dropped, or else return a corresponding error message.

Payload restrictions: - The name must be the name of a valid class

- The request must be one of “enroll”, “drop”

Endpoint: /places/class_history/

Methods: GET

Authorization Required: Yes

Fields/payload: None

Return value: Returns a list of all of the user’s classes, sorted by start date.

Description: This endpoint will return all of the user’s classes, sorted by the start date.

E.x [[“2022-10-17“, yoga”], [“2021-6-9“, zumba”]]

Payload restrictions: None

Endpoint: /places/filter_studios/

Methods: POST

Authorization Required: No

Fields/payload: type, value

Return value: Returns a list of all of the user’s classes, sorted by start date.

Description: This endpoint will return all of the studios, filtered by the type provided.

Payload restrictions: - type must be in “name”, “amenities”, “class_names”, and “coaches”

Endpoint: /places/filter_classes/

Methods: POST

Authorization Required: No

Fields/payload: studio_name, type, value

Return value: Returns a list of all of the user's classes, sorted by start date.

Description: This endpoint will return all of the classes of the given studio "studio_name", filtered by the type provided.

Payload restrictions: type must be in "class_name", "date", "time_range", and "coach_name"

The subs endpoint has been split between its 3 request types (GET, POST and DELETE)

Endpoint: /subs/

Method: GET

Authorization Required: No

Fields/payload: None

Return value: [{"amount": amount, "plan": plan}, ...] (on success)

Description: View all available subscriptions. A list of dictionaries will be returned, with each dictionary having its price (amount) and membership plan listed. The plan will be one of "Weekly", "Biweekly", "Monthly" and "Yearly".

With the interval between the installments depending on the subscription plan.

Payload restrictions: None

Endpoint: /subs/

Method: POST,

Authorization Required: Yes

Fields/payload: plan, amount

Return value: "Successfully updated subscription"(on success)

Description: Add the payload subscription plan to the member, or else edit it (replacing the previous subscription).

A new subscription will result in an automatic first payment being added to the the member's payment history, and future payments being added based on the subscription type.

Payload restrictions: - the plan and amount must match an available subscription (listed from the GET response)

- Plan must be one of "Weekly", "Biweekly", "Monthly" and "Yearly"
- Amount must be a float in the form xx.xx

Endpoint: /subs/

Methods: DELETE

Authorization Required: Yes

Fields/payload: None

Return value: "successfully deleted subscription" (on success)

"No subscription found" (if you try to delete without an existing subscription)

Description: Deletes the logged in member's subscription, if it exists. If the subscription is deleted then all future payment installations are canceled. Returns a message indicating success.

Restrictions: Member must have a subscription in order to delete it.

Endpoint: /places/studio/<studio_name>/update_amenities/

Methods: POST

Authorization Required: Yes, you must be admin

Fields/payload: type, quantity

Return value: “updated amenities <amenities>” (on success)

Description: Updates the amenities of the specified studio

Restrictions: Studio must exist, and you must be admin

Admin Panel - Notes

Create/edit studio: - location must be in the form (latitude, longitude) where $-90 \leq \text{latitude} \leq 90$ and $-180 \leq \text{longitude} \leq 180$

- The postal code can only contain 6 characters (excluding spaces)
- The phone number must be composed of 10 digits
- The images must be in .png or .jpg format
- amenities must be in the following format: type:quantity,type:quantity,type:quantity

Create/edit class: The time must be in the format 'Day|HH:MM|HH:MM|MMDDYY|MMDDYY' where day = Monday, Tuesday, ...

Create/edit subscription: The amount must be a positive float with no more than two (xx.xx) decimal places.