Labs Backend Challenge - Kaushik API Docs

GET Get All Clubs

http://127.0.0.1:5000/api/clubs

Returns all clubs within the system.

Example Request

Get All Clubs

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("GET", "/api/clubs", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

GET Get User By Username

http://127.0.0.1:5000/api/users/<username>

Takes input of `username`, a string, and returns that user's public profile

Example Request

Get User By Username

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("GET", "/api/users/<username>", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

GET Search Club

http://127.0.0.1:5000/api/clubs/search/<club_search_string>

Takes input of `club_search_substring` and returns all matching clubs that contain the string.
Case Insensitive!!

Example Request — Search Club

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("GET", "/api/clubs/search/<club_search_string>", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

POST Add Club

http://127.0.0.1:5000/api/clubs

Can provide either a list of clubs (shown below) or a single club in JSON format.
Will add clubs to the system.
Will skip clubs with a `code` that already exist within the system.
Will return all clubs added into the system / found already existing in the system with the same `club_code` in JSON format.

BODY raw

```
[
    {
    "name": "club5",
    "code": "code7",
    "description": "description5",
    "tags": ["Athletics", "Undergraduate"]
    },
    {
    "name": "club6",
    "code": "code8",
```

View More

Example Request

Add Club

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "[\r\n    {\r\n    \"name\": \"club5\",\r\n    \"code\": \"code7\",\r\n    \"description\":
headers = {}
conn.request("POST", "/api/clubs", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

PUT Add Favorite

http://127.0.0.1:5000/api/users/<username>/favorites

Uses the `username` URL identifier and the JSON body which should provide the `club_code` for the club that the user in question should have favorited (example body below).
If a club is already within a user's favorites, it will remove this item from their favorites list (serves as a sort of toggle function for favorites)

BODY raw

```
{
    "club_code": "locustlabs"
}
```

Example Request

Add Favorite

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"club_code\": \"locustlabs\"\r\n}"
headers = {}
conn.request("PUT", "/api/users/<username>/favorites", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

PUT Modify Club

http://127.0.0.1:5000/api/clubs

Takes input of a club's information in JSON format (example below).
Only required field is `code`. Will change only the data provided by the user in this JSON and leaves the remaining fields unchanged.
If club isn't found, will return with an `HTTP.NOT_FOUND (404)` error. Needs a valid, existing club code.
Can only modify one club at a time, will return with an `HTTP.METHOD_NOT_ALLOWED (405)` error if multiple clubs are passed.
Will return the modified club information in JSON format.

BODY raw

```
{
    "name": "club4 mod",
    "code": "code7",
    "description": "description4 modkeepold",
    "tags": ["Undergraduate"]
}
```

Example Request Modify Club

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"name\": \"club4 mod\",\r\n    \"code\": \"code7\",\r\n    \"description\": \"des
headers = {}
conn.request("PUT", "/api/clubs", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

GET Get All Tags

http://127.0.0.1:5000/api/clubs/tags

Returns all tags that clubs are classified with in the system along with the number of clubs that have each tag assigned to them.

Example Request

Get All Tags

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("GET", "/api/clubs/tags", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

GET Get All Users

http://127.0.0.1:5000/api/users

Returns all users' data from the system, including identifying elements for comments and favorite clubs.

Example Request

Get All Users

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("GET", "/api/users", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

POST Add User

http://127.0.0.1:5000/api/users

Will add a user given user data in JSON format (example below).
If username already exists within system, that user is simply returned and no changes are made to the data storage.
All fields are necessary, or will return with `HTTP.NOT_FOUND (404)` error.
Will return the newly added user (or the user with the already existing username) in JSON format with id information.

BODY raw

```
{
    "name": "Josh2",
    "username": "josh2",
    "email": "josh2@upenn.edu",
    "gender": 1,
    "year": "sophomore"
}
```

Example Request — Add User

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"name\": \"Josh2\",\r\n    \"username\": \"josh2\",\r\n    \"email\": \"josh2@upe
headers = {}
conn.request("POST", "/api/users", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

POST Add Comment

http://127.0.0.1:5000/api/users/comment

Will add a user comment given user, club, and comment data in JSON format (example below).
It is possible for a user to comment multiple times for the same club.
All fields are necessary, or will return with `HTTP.NOT_FOUND (404)` error.

All fields are necessary, or will return with `HTTP.NOT_FOUND (404)` error.
Will return the added comment in JSON format with id information.

Documentation Settings

BODY raw

```
{
    "user_id": 1,
    "club_id": 1,
    "text": "testcomment 1 Hello"
}
```

Example Request

Add Comment

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"user_id\": 1,\r\n    \"club_id\": 1,\r\n    \"text\": \"testcomment 1 Hello\"\r\
headers = {}
conn.request("POST", "/api/users/comment", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

PUT Modify Comment

http://127.0.0.1:5000/api/users/comment

Given a `comment_id`, an integer value, and `text`, a string value in JSON format (example below) if the comment is found the text will be replaced within the comment, leaving all other data unchanged.
If comment_id isn't provided or if the requested comment isn't found in the system, will return with `HTTP.NOT_FOUND (404)` error specifying which issue it is
Returns all fields of the modified comment in JSON format.

BODY raw

```
{
    "comment_id": "1",
    "text": "testcomment 1 Hello modified"
}
```

Example Request

Modify Comment

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"comment_id\": \"1\",\r\n    \"text\": \"testcomment 1 Hello modified\"\r\n}"
headers = {}
conn.request("PUT", "/api/users/comment", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

DEL Delete Comment

http://127.0.0.1:5000/api/users/comment

Given a `comment_id`, an integer value in JSON format (example below), the comment alongside associations with the respective club and user will be deleted.
If `comment_id` isn't provided or if the requested comment isn't found in the system, will return with `HTTP.NOT_FOUND (404)` error specifying which issue it is
Returns all fields of the deleted comment in JSON format.

BODY raw

```
{
    "comment_id": 1
}
```

Example Request

Delete Comment

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = "{\r\n    \"comment_id\": 1\r\n}"
headers = {}
conn.request("DELETE", "/api/users/comment", payload, headers)
res = conn.getresponse()
data = res.read()
print(data.decode("utf-8"))
```

POST Webscrape

POST Webscrape

Documentation Settings

http://127.0.0.1:5000/api/clubs/webscrape

Takes no input, but will scrape clubs from the follwing url: https://ocwp.pennlabs.org/ (https://ocwp.pennlabs.org/)
Scraped clubs are then assigned a pseudo-random code and added into the storage system
Returns all of the clubs added into the system in JSON format.

Example Request

Webscrape

```
import http.client

conn = http.client.HTTPSConnection("127.0.0.1", 5000)
payload = ''
headers = {}
conn.request("POST", "/api/clubs/webscrape", payload, headers)
res = conn.getresponse()
data = res.read()
```