

CPSC 449 - Fall 2019 - Project 2

Names

Oscar Olazabal
Shivam Shah
Kevin Chen

Dependencies

This project depends on Foreman and all of the Python modules listed in `requirements.txt`. To install foreman and the modules and run:

```
sudo apt install ruby-foreman
pip3 install -r requirements.txt
```

You will also need to install [Kong](#) and [MinIO](#).

Follow Step 1 of the Ubuntu Installation instructions to install Kong on Tuffix using the .deb package for Ubuntu 18.04 Bionic. Then use the following commands to install PostgreSQL and create a database and configuration file for Kong:

```
sudo apt install --yes postgresql
sudo -u postgres psql -c "CREATE USER kong WITH ENCRYPTED PASSWORD 'kong'"
sudo -u postgres psql -c 'CREATE DATABASE kong OWNER kong'
sudo cp /etc/kong/kong.conf.default /etc/kong/kong.conf
```

Now edit `/etc/kong/kong.conf` to uncomment `pg_password` and change it to kong, then use the following commands to start Kong:

```
sudo kong migrations bootstrap
ulimit -n 4096 && sudo kong start
```

You can then proceed with the curl command in Step 4 of the Ubuntu Installation instructions to verify that Kong is running.

For Minio, create a new directory to store the data for the server, and follow the directions on their website to start the MinIO server. Be sure to start the server with the command `./minio server buckets/`

When the MinIO server starts, it will print an endpoint, access key, and secret key. Make a note of these values so that you can access the server later.

Use the information provided to log into the MinIO Browser and create a bucket named tracks. Edit the policy for this bucket to add the prefix `*` as Read Only. Upload some music files and verify that you can access them via URLs such as <http://localhost:9000/tracks/music-file.mp3>

If you run into dependency errors, ensure all of the modules in `requirements.txt` are install properly. You may have to use `apt install` to install some.

Running the Application

To run the application, you have to initialize the database and run each service on their own port. To do so, run the following:

```
./start
```

Testing the Application

To test the application you can use the provided `test` executable.

```
./test
```

The executable creates, retrieves, updates, and deletes from the database using the endpoints of the application

Flask Output

```
student@tuffix-vm:~/Desktop/csu449-projects/project2$ ./test

INSERTING NEW USER
18:28:00 users.2 | 127.0.0.1 - - [27/Nov/2019 18:28:00] "POST /users HTTP/1.1" 201 -
(b'{"username": "username1", "user_pass": "pbkdf2:sha256:150000$cCiNZZMr$3d316c'
b'bc8e01db22f12ba80d120008e198966b1c2793d2e2a3ef7465e706bd19", "disp_name": "i'
b'amuser1", "email": "user1@gmail.com", "url_homepage": "user1.com", "id": 1}')

INSERTING NEW USER
18:28:01 users.2 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "POST /users HTTP/1.1" 201 -
(b'{"username": "username2", "user_pass": "pbkdf2:sha256:150000$q7xsWoLr$a2c36c'
b'6544275e782c360e006c5051cae9e709557140a6a56b96fb2eef7cb83e", "disp_name": "i'
b'amuser2", "email": "user2@gmail.com", "url_homepage": "user2.com", "id": 2}')

INSERTING NEW USER
18:28:01 users.2 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "POST /users HTTP/1.1" 201 -
(b'{"username": "username3", "user_pass": "pbkdf2:sha256:150000$30LdGEKq$c7ebf1'
b'a96b7c90649d653b493105204eacda221cf0491c51138135c0bf39e155", "disp_name": "i'
b'amuser3", "email": "user3@gmail.com", "url_homepage": "user3.com", "id": 3}')

GET USER
18:28:01 users.1 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "GET /users/username1 HTTP/1.1" 200 -
(b'{"username": "username1", "disp_name": "iamuser1", "email": "user1@gmail.com'
b'", "url_homepage": "user1.com"}'})

GET USER
18:28:01 users.1 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "GET /users/username2 HTTP/1.1" 200 -
(b'{"username": "username2", "disp_name": "iamuser2", "email": "user2@gmail.com'
b'", "url_homepage": "user2.com"}'})

GET USER
18:28:01 users.1 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "GET /users/username3 HTTP/1.1" 200 -
(b'{"username": "username3", "disp_name": "iamuser3", "email": "user3@gmail.com'
b'", "url_homepage": "user3.com"}'})

DELETE USER
18:28:01 users.1 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "DELETE /users/username3 HTTP/1.1" 200 -
b'{"message": "Deleted 1 user with username username3"}'

UPDATE USER PASSWORD
18:28:01 users.2 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "PATCH /users/username1 HTTP/1.1" 200 -
(b'{"username": "username1", "disp_name": "iamuser1", "email": "user1@gmail.com'
b'", "url_homepage": "user1.com"}'})

AUTHENTICATE USER
18:28:01 users.1 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "GET /users/authenticate HTTP/1.1" 200 -
b'{"result": true}'

NEW TRACK
18:28:01 tracks.3 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "POST /tracks HTTP/1.1" 201 -
{'album_art_url': 'https://picsum.photos/id/975/200/200',
```

```
'album_title': 'rock',
'artist': 'Flea',
'id': '02754400-bccb-4e07-bf0c-527166a70ee8',
'media_file_url': 'http://localhost:8000/media/bass.mp3',
'track_length': '2.1',
'track_title': 'bass'}
```

NEW TRACK

```
18:28:01 tracks.3 | 127.0.0.1 - - [27/Nov/2019 18:28:01] "POST /tracks HTTP/1.1" 201 -
{'album_art_url': 'https://picsum.photos/id/876/200/200',
'album_title': 'rock',
'artist': 'Slash',
'id': 'e017fec2-b32c-4f13-bc7c-3e38fd570f98',
'media_file_url': 'http://localhost:8000/media/electric-guitar.mp3',
'track_length': '2.1',
'track_title': 'electric guitar'}
```

NEW TRACK

```
18:28:02 tracks.1 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /tracks HTTP/1.1" 201 -
{'album_art_url': 'https://picsum.photos/id/1062/200/200',
'album_title': 'classy',
'artist': 'Yoyoma',
'id': 'd24ee1a3-37f2-4ba1-bdf8-205e3471c3fc',
'media_file_url': 'http://localhost:8000/media/harp.mp3',
'track_length': '2.1',
'track_title': 'harp'}
```

UPDATE TRACK

```
18:28:02 tracks.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "PATCH /tracks/d24ee1a3-37f2-4ba1-bdf8-
205e3471c3fc HTTP/1.1" 200 -
{'album_art_url': 'https://picsum.photos/id/1062/200/200',
'album_title': 'classier',
'artist': 'Yoyoma',
'media_file_url': 'http://localhost:8000/media/violin.mp3',
'track_length': 3.1,
'track_title': 'violin',
'uuid': 'd24ee1a3-37f2-4ba1-bdf8-205e3471c3fc'}
```

GET TRACK

```
18:28:02 tracks.1 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /tracks/d24ee1a3-37f2-4ba1-bdf8-
205e3471c3fc HTTP/1.1" 200 -
{'album_art_url': 'https://picsum.photos/id/1062/200/200',
'album_title': 'classier',
'artist': 'Yoyoma',
'media_file_url': 'http://localhost:8000/media/violin.mp3',
'track_length': 3.1,
'track_title': 'violin',
'uuid': 'd24ee1a3-37f2-4ba1-bdf8-205e3471c3fc'}
```

DELETE TRACK

```
18:28:02 tracks.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "DELETE /tracks/d24ee1a3-37f2-4ba1-bdf8-
205e3471c3fc HTTP/1.1" 200 -
{'message': 'Deleted 1 track with id d24ee1a3-37f2-4ba1-bdf8-205e3471c3fc'}
```

NEW DESCRIPTION

```
18:28:02 descriptions.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /descriptions HTTP/1.1" 201 -
{'description': 'this is a track description',
'id': '02754400-bccb-4e07-bf0c-527166a70ee8',
'user_name': 'username1'}
```

NEW DESCRIPTION

```
18:28:02 descriptions.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /descriptions HTTP/1.1" 201 -
{'description': 'this is a track description again',
'id': 'e017fec2-b32c-4f13-bc7c-3e38fd570f98',
'user_name': 'username2'}
```

GET DESCRIPTION

```
18:28:02 descriptions.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /descriptions/02754400-bccb-4e07-
bf0c-527166a70ee8 HTTP/1.1" 200 -
{'description': 'this is a track description',
'id': '02754400-bccb-4e07-bf0c-527166a70ee8',
```

```
'user_name': 'username1'}

INSERTING NEW PLAYLIST
18:28:02 playlists.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /playlists HTTP/1.1" 201 -
{'creator': 'username1',
 'description': 'this is playlist1',
 'id': 1,
 'title': 'playlist1',
 'tracks': ['02754400-bccb-4e07-bf0c-527166a70ee8']}

INSERTING NEW PLAYLIST
18:28:02 playlists.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /playlists HTTP/1.1" 201 -
{'creator': 'username2',
 'description': 'this is playlist2',
 'id': 2,
 'title': 'playlist2',
 'tracks': ['02754400-bccb-4e07-bf0c-527166a70ee8',
            'e017fec2-b32c-4f13-bc7c-3e38fd570f98']}

INSERTING NEW PLAYLIST
18:28:02 playlists.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "POST /playlists HTTP/1.1" 201 -
{'creator': 'username2',
 'description': 'this is playlist3 made by username2',
 'id': 3,
 'title': 'playlist3',
 'tracks': ['e017fec2-b32c-4f13-bc7c-3e38fd570f98']}

GET ALL PLAYLISTS
18:28:02 playlists.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /playlists/all HTTP/1.1" 200 -
[{'creator': 'username1',
  'description': 'this is playlist1',
  'id': 1,
  'title': 'playlist1',
  'tracks': ['02754400-bccb-4e07-bf0c-527166a70ee8']},
 {'creator': 'username2',
  'description': 'this is playlist2',
  'id': 2,
  'title': 'playlist2',
  'tracks': ['02754400-bccb-4e07-bf0c-527166a70ee8',
            'e017fec2-b32c-4f13-bc7c-3e38fd570f98']},
 {'creator': 'username2',
  'description': 'this is playlist3 made by username2',
  'id': 3,
  'title': 'playlist3',
  'tracks': ['e017fec2-b32c-4f13-bc7c-3e38fd570f98']}]

GET PLAYLIST
18:28:02 playlists.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /playlists/1 HTTP/1.1" 200 -
{'creator': 'username1',
 'description': 'this is playlist1',
 'id': 1,
 'title': 'playlist1',
 'track': ['02754400-bccb-4e07-bf0c-527166a70ee8']}

GET PLAYLIST
18:28:02 playlists.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /playlists/2 HTTP/1.1" 200 -
{'creator': 'username2',
 'description': 'this is playlist2',
 'id': 2,
 'title': 'playlist2',
 'track': ['02754400-bccb-4e07-bf0c-527166a70ee8',
            'e017fec2-b32c-4f13-bc7c-3e38fd570f98']}

GET PLAYLIST
18:28:02 playlists.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /playlists/3 HTTP/1.1" 200 -
{'creator': 'username2',
 'description': 'this is playlist3 made by username2',
 'id': 3,
 'title': 'playlist3',
 'track': ['e017fec2-b32c-4f13-bc7c-3e38fd570f98']}
```

GET PLAYLIST BY CREATOR

```
18:28:02 playlists.3 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "GET /playlists?creator=username1 HTTP/1.1" 200 -  
[{'creator': 'username1',  
  'description': 'this is playlist1',  
  'id': 1,  
  'title': 'playlist1',  
  'tracks': ['02754400-bccb-4e07-bf0c-527166a70ee8']}]
```

DELETE PLAYLIST

```
18:28:02 playlists.2 | 127.0.0.1 - - [27/Nov/2019 18:28:02] "DELETE /playlists/3 HTTP/1.1" 200 -  
{'message': 'Deleted 1 playlist with id 3'}
```

Endpoints

Tracks **PORT 8000**

- `/tracks/<uuid:id>`
 - **GET** Retrieves a track given its UUID
 - **DELETE** Deletes a track given its UUID
 - **PATCH** Updates a track given its UUID and the JSON data of the fields to be modified
- `/tracks`
 - **POST** Inserts a track given the JSON data of a new track

Playlists **PORT 8000**

- `/playlists/<int:id>`
 - **GET** Retrieves a playlist given its ID
 - **DELETE** Deletes a playlist given its ID
- `/playlists`
 - **GET** Retrieves a playlist of a user given the argument `creator=`
 - **POST** Inserts a playlist given the JSON data of a new playlist
- `/playlists/all`
 - **GET** Retrieves all playlists

Users **PORT 8000**

- `/users/<username>`
 - **GET** Retrieves user's profile (does not include password) given their username
 - **PATCH** Updates a user's password, given an updated password in JSON format
 - **DELETE** Deletes a user given their username
- `/users`
 - **POST** Inserts a new user given their data in JSON format
- `/users/authenticate`
 - **GET** Validates a user and password given JSON data containing a username and password

Descriptions **PORT 8000**

- `/descriptions/<uuid:id>`
 - **GET** Retrieves user descriptions given the UUID of a track
- `/descriptions`
 - **POST** Inserts a user description of a track given a username, a track UUID, and a description

XSPF **PORT 8000**

- `/playlist/<int:id>.xspf`
 - GET Retrieves playlist given its ID and renders a XSPF of it's information. If you have the correct software installed (such as Parole Media Player), you will be prompted to open the playlist and stream the music

Data Structure

Below are sample JSONs of the entities.

Tracks

```
{
  "id" : UUID
  "track_title": "Track Title",
  "album_title": "Album Title",
  "artist": "Artist's Name",
  "track_length": 2.6,
  "media_file_url": "file:///home/student/Music/mediafile1",
  "album_art_url": "file:///home/student/Music/albumart1"
}
```

Playlists

```
{
  "title": "Playlist Title",
  "creator": "Username",
  "tracks" : [
    "file:///home/student/Music/media_file1",
    "file:///home/student/Music/media_file2"
  ],
  "description": "Description of playlist"
}
```

Users

```
{
  "username" : "User name",
  "user_pass" : "Password of user",
  "disp_name" : "Display name",
  "email" : "useremail@email.com",
  "url_homepage" : "http://userhomepage.com"
}
```

Descriptions

```
{
  "track_id": UUID,
  "user_name": "User name",
  "description": "This is a track description"
}
```