

Onur Ozay

CS 419 - Mobile/Cloud Software Development - Final Project

August 12, 2016

The video can be seen at: <https://www.youtube.com/watch?v=YtVXILDRxls>

## Introduction

For the final project I created an application that shows the current location of the user. The initial Sign-In page asks user to log in (or sign up) using a username and password. Once the user enters a username and password and clicks on the next button, the app takes the user to the Current Location page where it greets the user with their name and displays the current location of the user in longitude and latitude format. On the Current Location page, the user has the option to save his/her current location by clicking on the save button. If the save button is clicked, the app will take the user to Saved Location page where the list of locations that have been visited is displayed. Also on the Saved Location page the user has the option to delete a specific location from the database by clicking the delete button next to each saved location.

I used React-Native for the front-end and I made an Android app. I also used Genymotion simulator for the Android phone. For the back-end, I used Node.JS, Express Framework and MongoDB database. I also used Mongoose which is a Node.js library that provides MongoDB object mapping -- It basically translates data in the database to JavaScript object.

I did not use any OAuth authorization system, I created a basic account system where the app only asks the user to create a username and password.

I used three HTTP verbs: GET, POST and DELETE.

## POST Request:

In order to create a new account, POST request is made to the server. The POST

request can be made at this :

URL => 54.191.149.78:3000/api/users/signin

A valid username and password are the required fields to make this request. Otherwise it will give an error message. Once the username and password entered in the body, it creates a new account for the new user and returns a response as below:

```
{
  "__v": 0,
  "username": "Phelps",
  "password": "phelps"
  "_id": "57ae2f91a5e8a2733d7f5c20",
  "createdAt": "2016-08-12T20:20:33.351Z",
  "locations": []
}
```

As seen above, the Locations and the Created Date are not required fields to create an account. Also for this request, I did not specify the location latitude and longitude and it shows as an empty list.

#### GET Method:

Once the account is created, the user info can be reach at the following:

URL => 54.191.149.78:3000/api/users/<username>

For example when I make a request to the server with the username Daenerys, the JSON formatted response would be:

```
{
  "_id": "57acd357a5e8a2733d7f5c0b",
  "username": "Daenerys",
  "password": "daenerys",
}
```

```

    "__v": 11,
    "createdAt": "2016-08-11T19:34:47.200Z",
    "locations": [
      {
        "latitude": "65.9667",
        "longitude": "-18.5333",
        "_id": "57acd457a5e8a2733d7f5c1d"
      }
    ]
  }
}

```

It automatically creates an ID number for each user, as well as an id number for each location that is visited or saved.

#### DELETE METHOD:

The delete method deletes the location selected by the user. It makes a delete request to the server with the following:

URL => 54.191.149.78:3000/api/users/<username>

In the body of the request, the location of the ID number should be listed. For example, if we make a request with the user name Daenerys and the location ID number "57acd457a5e8a2733d7f5c1d", it would delete that particular location for the user. The JSON formatted response would be:

```

{
  "_id": "57acd357a5e8a2733d7f5c0b",
  "username": "Daenerys",
  "password": "daenerys",
  "__v": 12,
  "createdAt": "2016-08-11T19:34:47.200Z",

```

```
"locations": []  
}
```

Screen Shot:

Below screen shots show how the API requests are made using Postman app.



