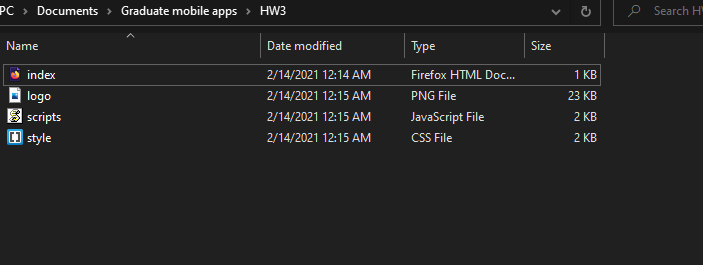
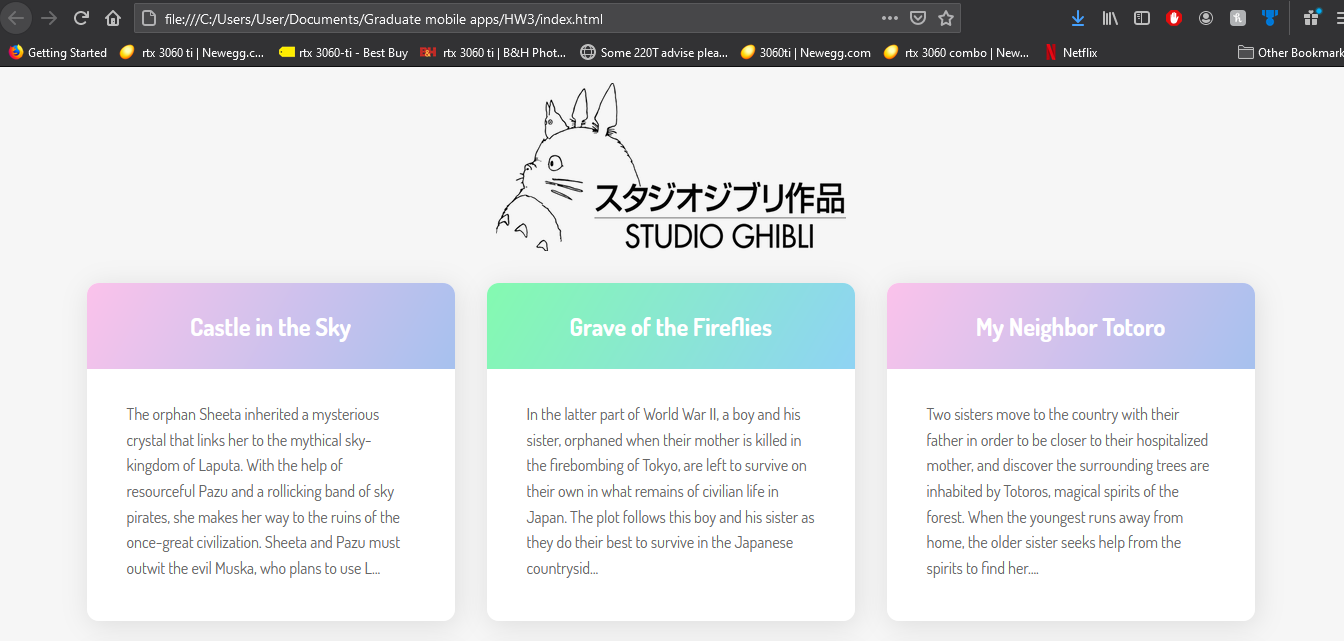
**Maaz Malik Z23385841**  **Mobile Apps Homework 3**

**2/15/2021 Prof. Jaramillo**

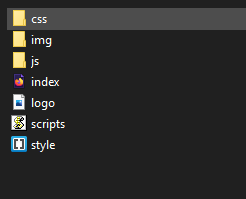
**1.  How to Connect to an API with JavaScript**

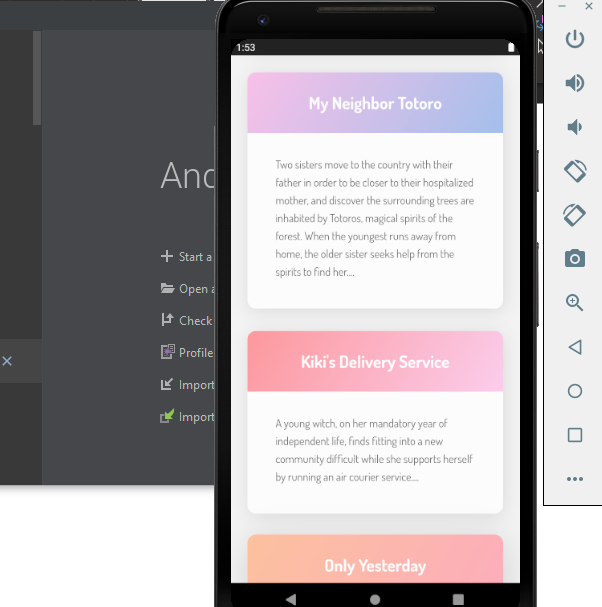


Here we have the files setup from the tutorial



and I got it to work on my Local Device just like the steps told me.

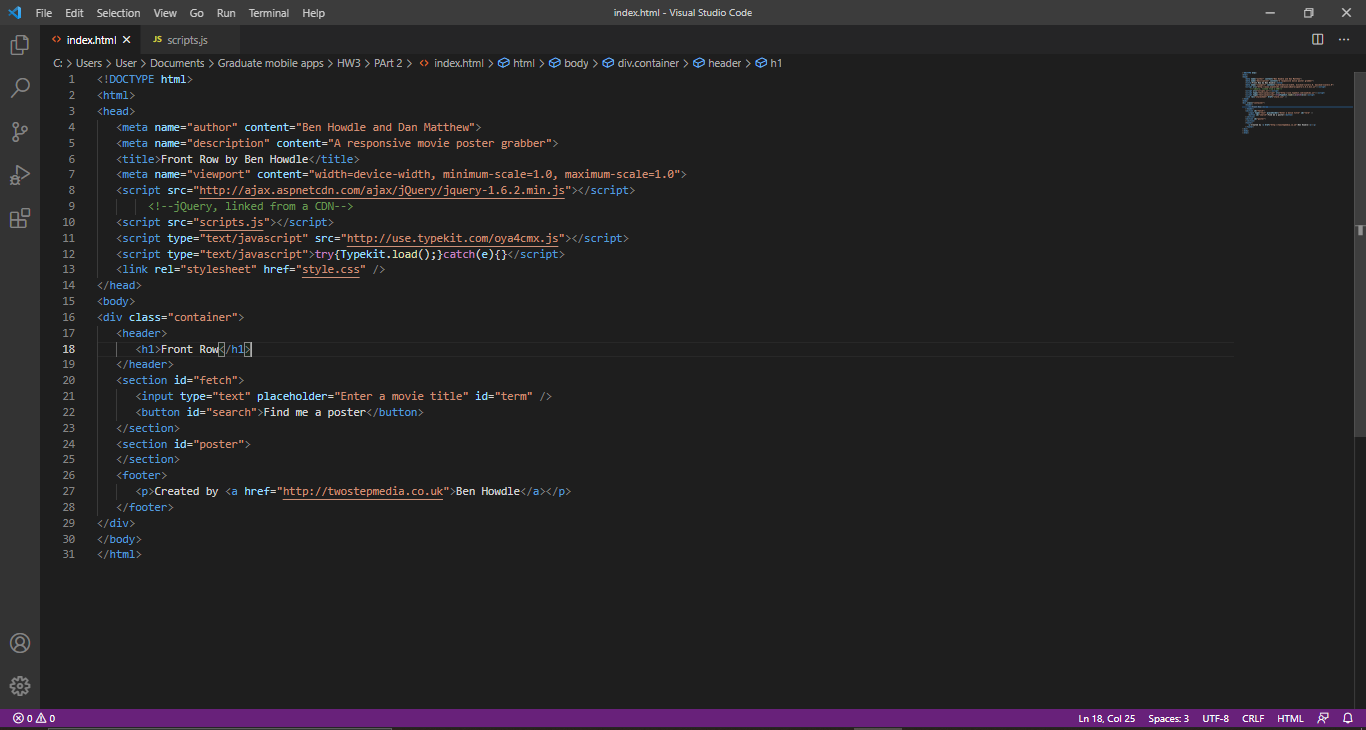
After a lot of trouble, I was able to put all the files into A created Cordova Project



And launched the App successfully on the Emulator through the AVD Manager. (Side note, My Neighbor Totoro is one of My Favorite Movies)

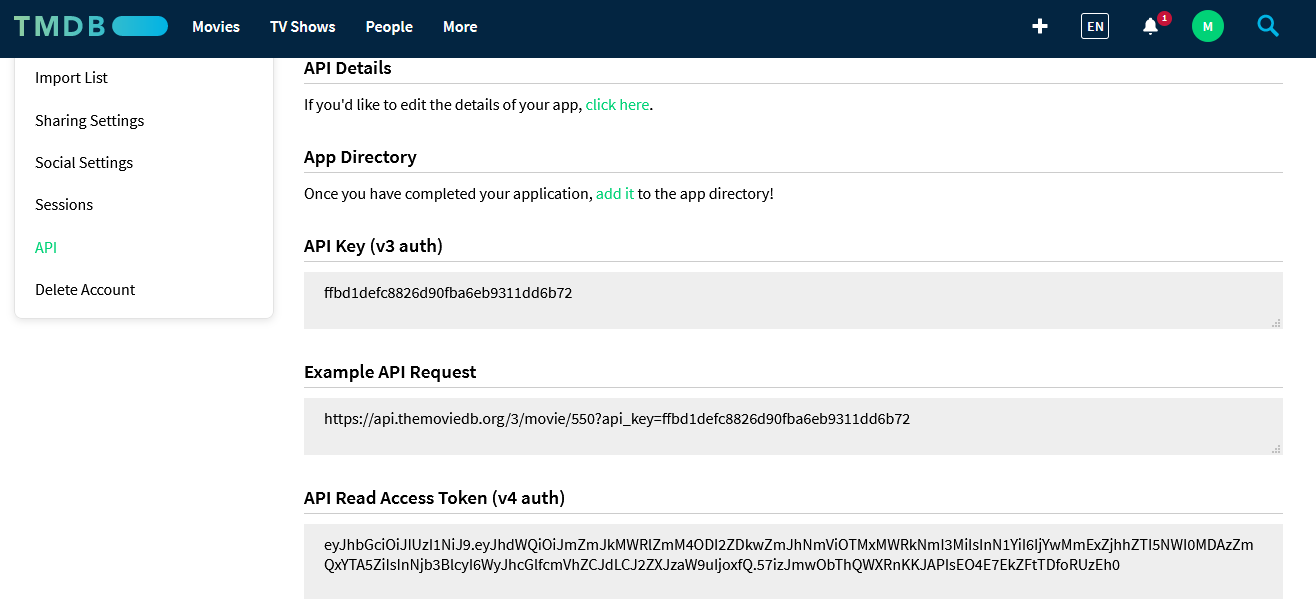
**2.  A Beginner’s Guide To jQuery-Based JSON API Clients**

So I started out with the Guide and followed along with the steps for Creating the index.html file and Script.js file



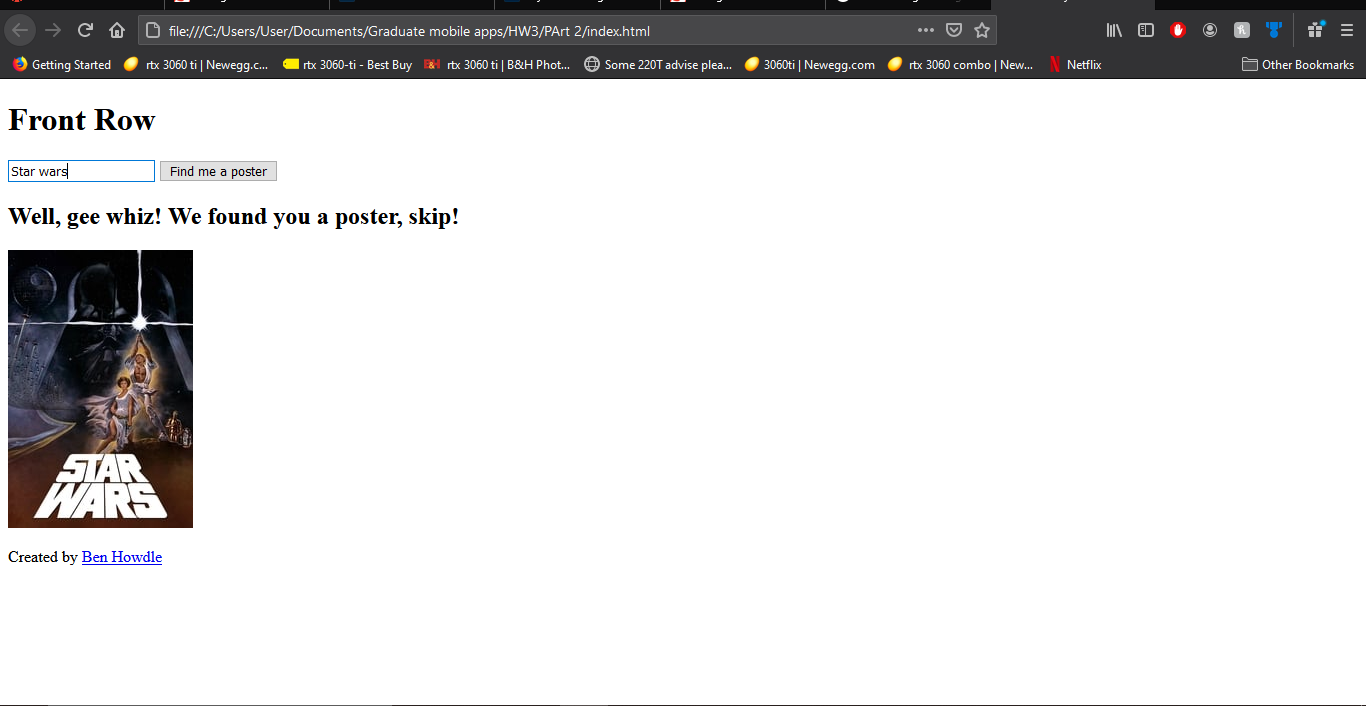
Originally I launched it as a test on the browser but the App was unresponsive. I looked at the Announcement and the Slack group and saw the updated code for the Script.js in the code bin. I Implemented that instead into the .js file.

I also had to go on the website and create an Account in order to get the API.

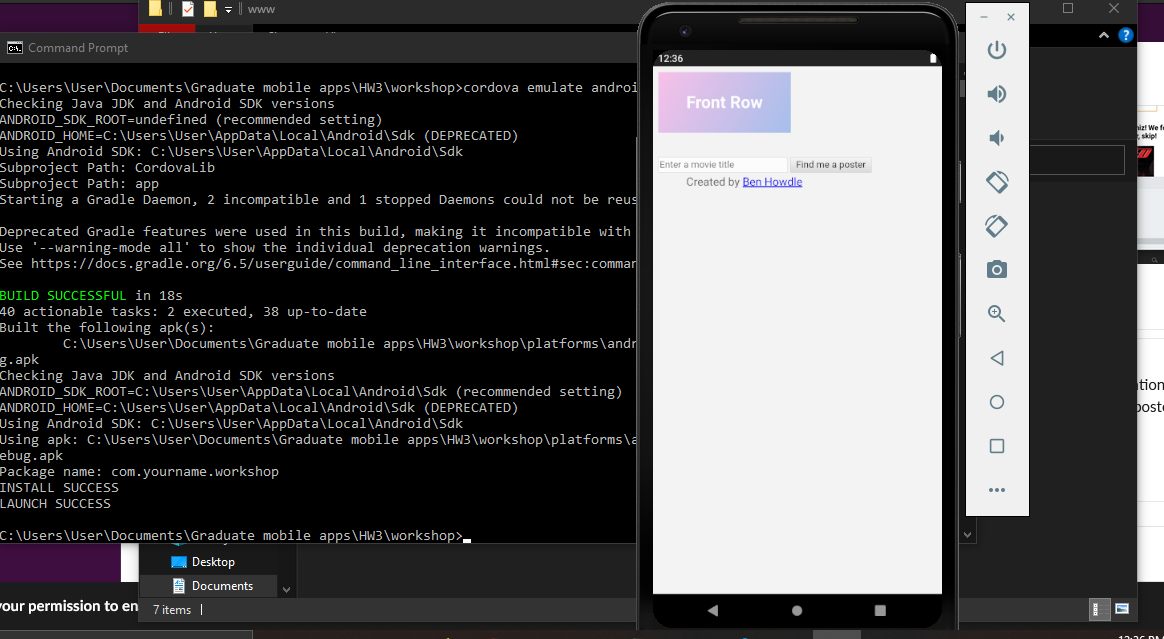


After a very unique setup process, I got my own API and pasted it into the sample code.

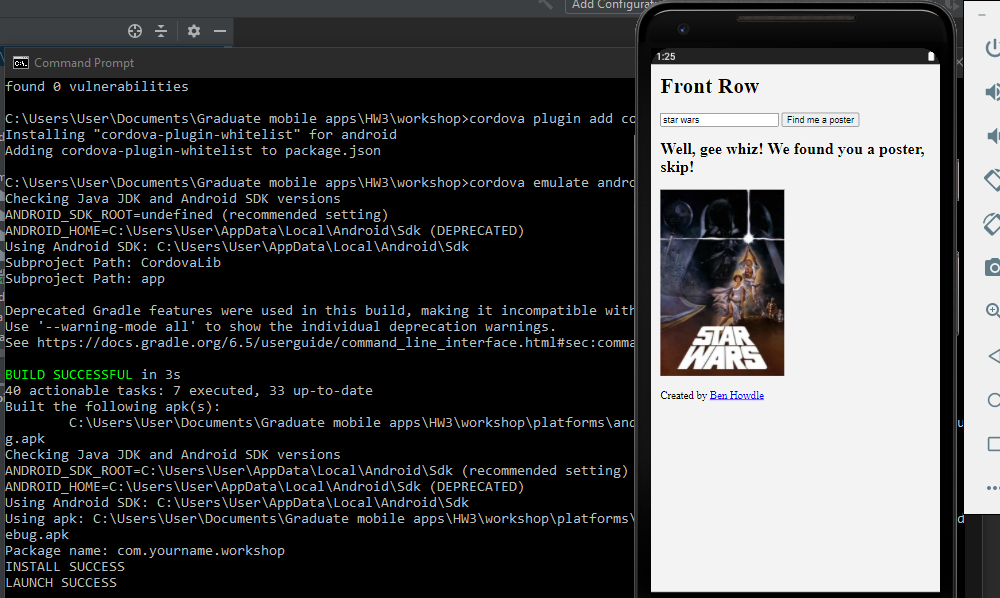
var API\_KEY = 'ffbd1defc8826d90fba6eb9311dd6b72';



And here was the result when I launched the Browser test

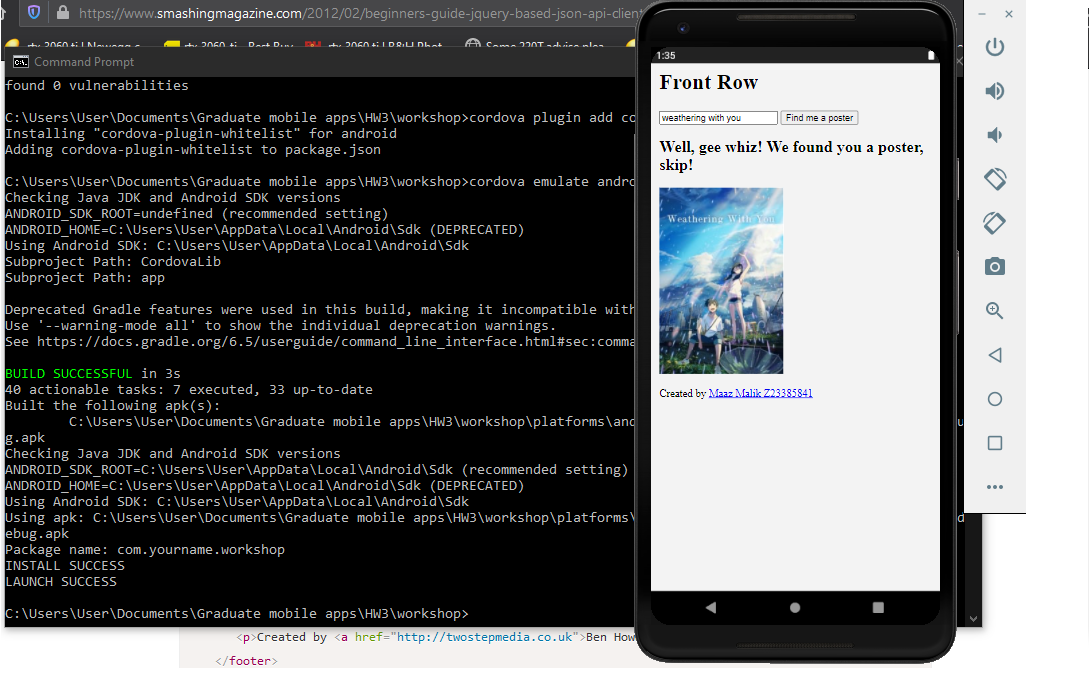


The First Android test launched successfully on the emulator. I didn’t run into any of the white page issues anyone else was having so far so good.



So initially I had the issue that was stated in the slack. I tried installing the whitelist but it was already installed. I then realized that my CSS was completely different from the Browser test I did earlier.

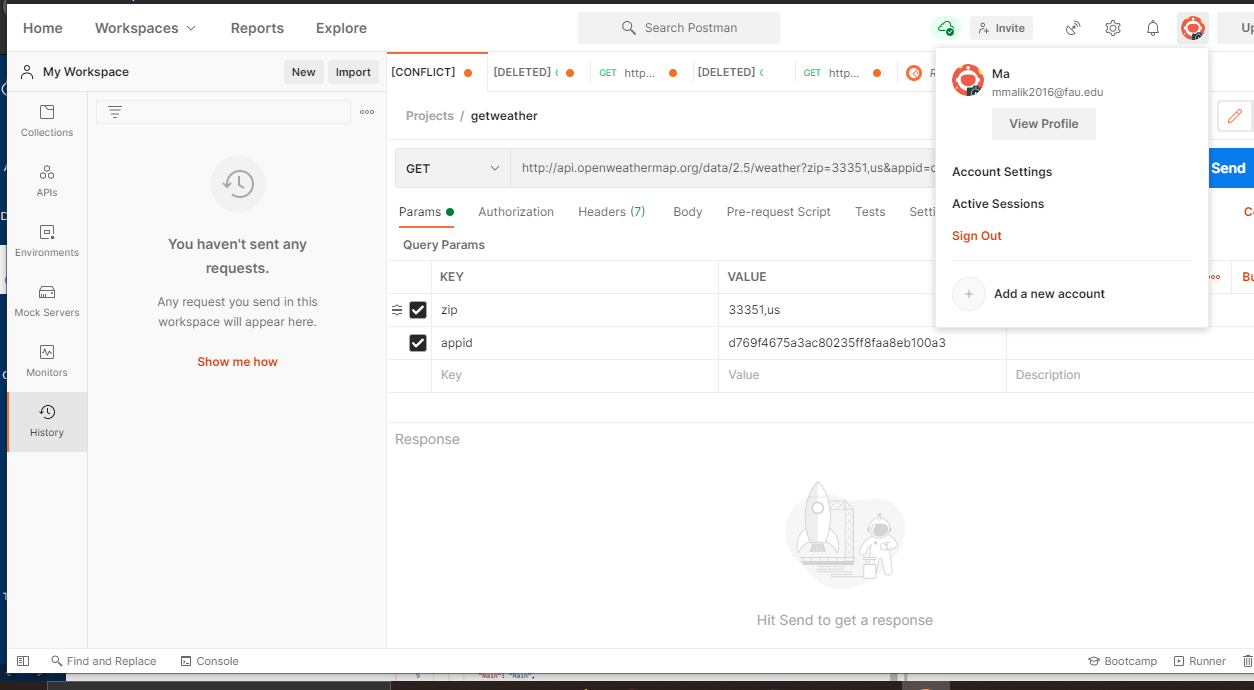
I realized that I had not properly directed the updated script.js file because I copied all these created fields into the already created Cordova project file from Part 1 so that why the CSS looked strange. I completely wiped it and pasted the Part 2 contents in the Cordova folder and Got it to work.



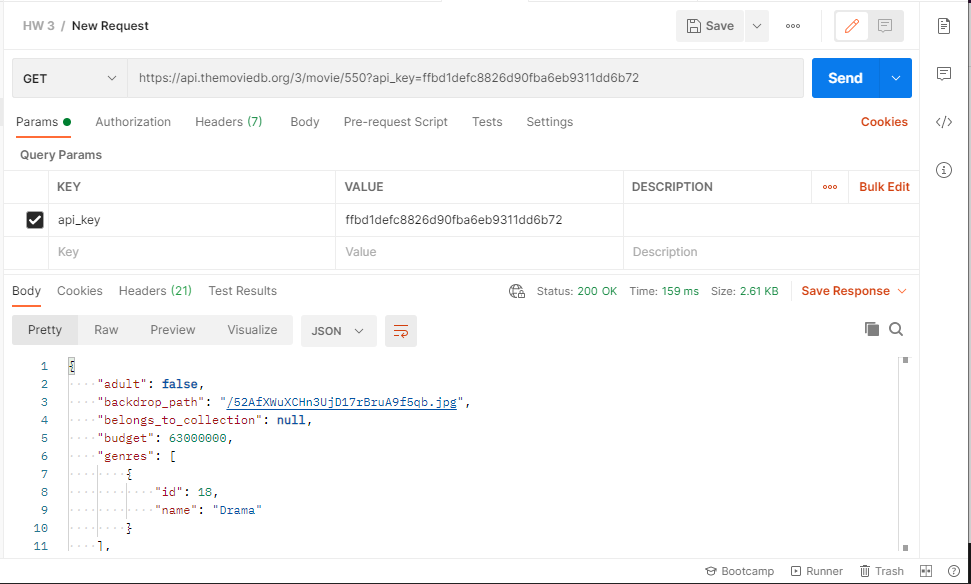
I then went ahead and added my name and Z number since the original creators was already on there.

**3 -Test Web Service APIs using Postman**

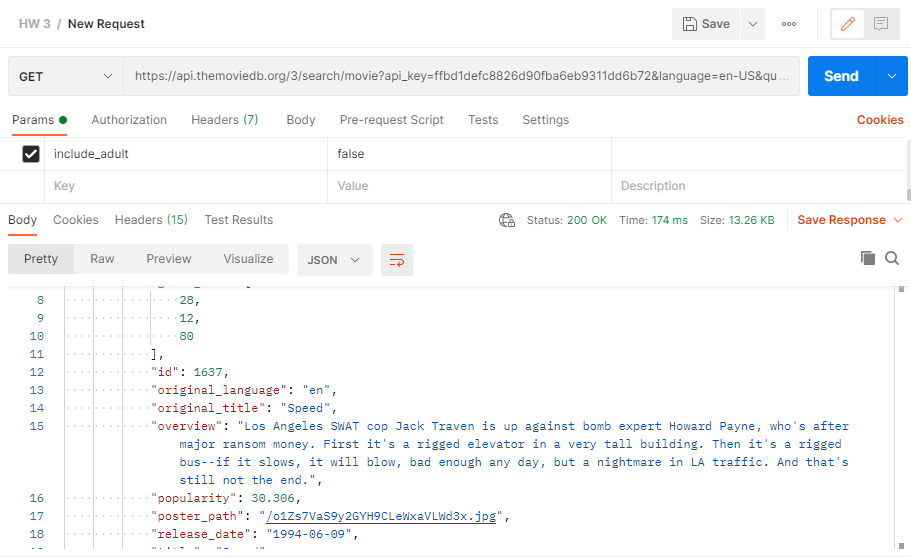
**a)**



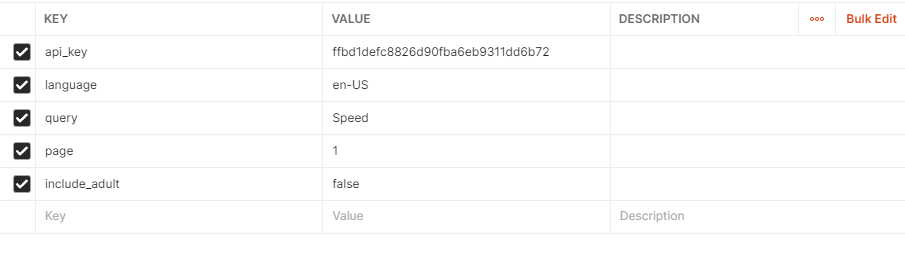
I already had Postman installed so I just had to download the installer in order to update it since the automatic updates weren’t working.



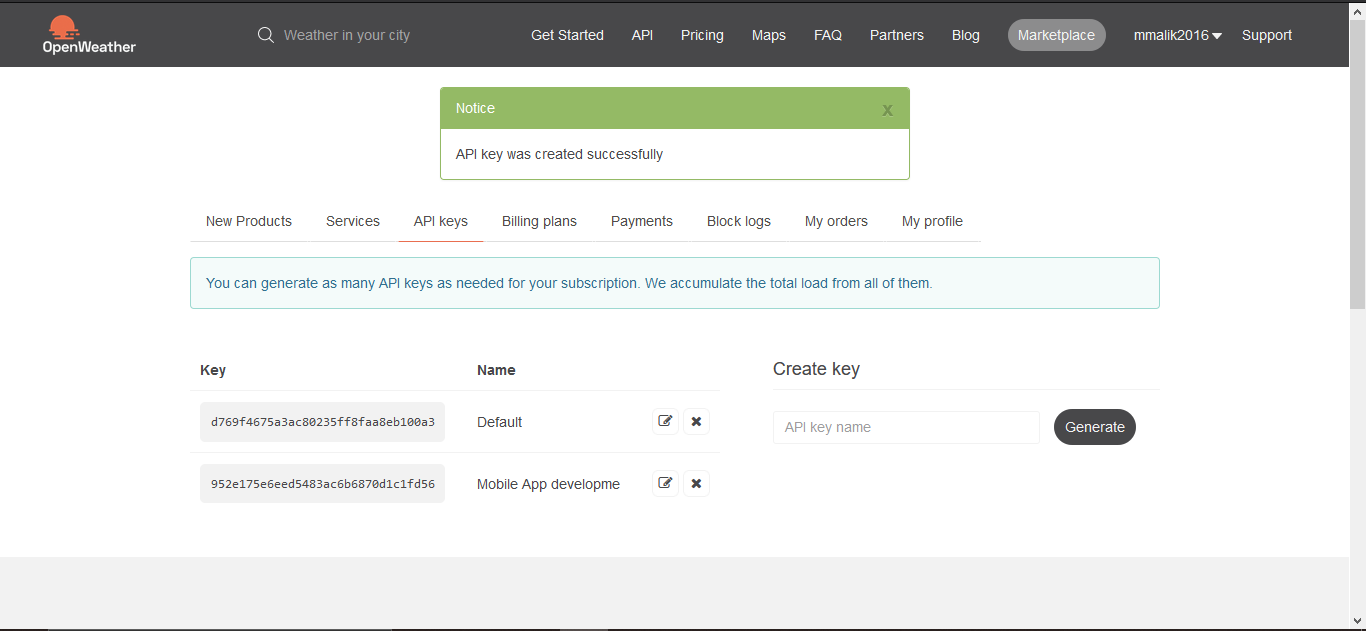
I created a new HW3 project folder and tested the API Pull request from the Movie app I did in Part 2. The Request successfully went through, and I got all the data displayed.



After struggling for a while I got some help from the TA on how to change my pull request. I was able to pull information on the Movie Speed using the API.



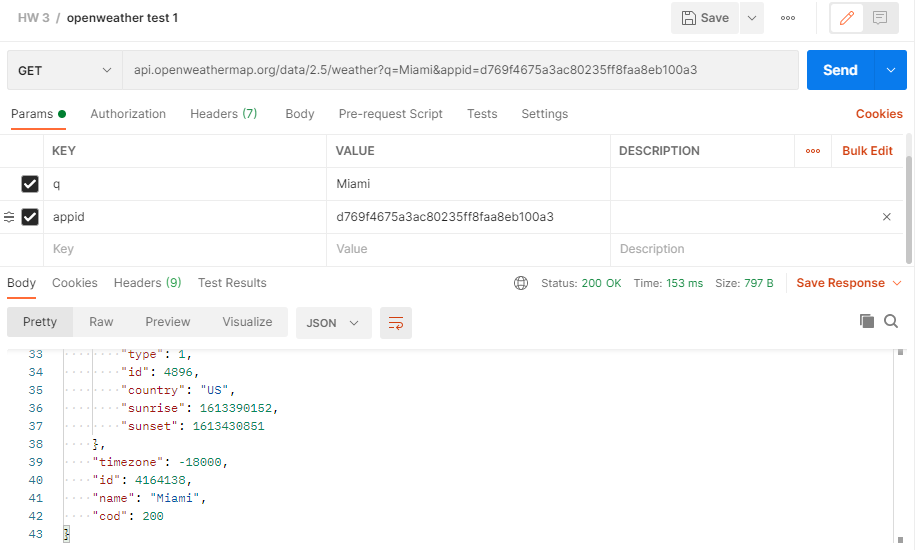
b)



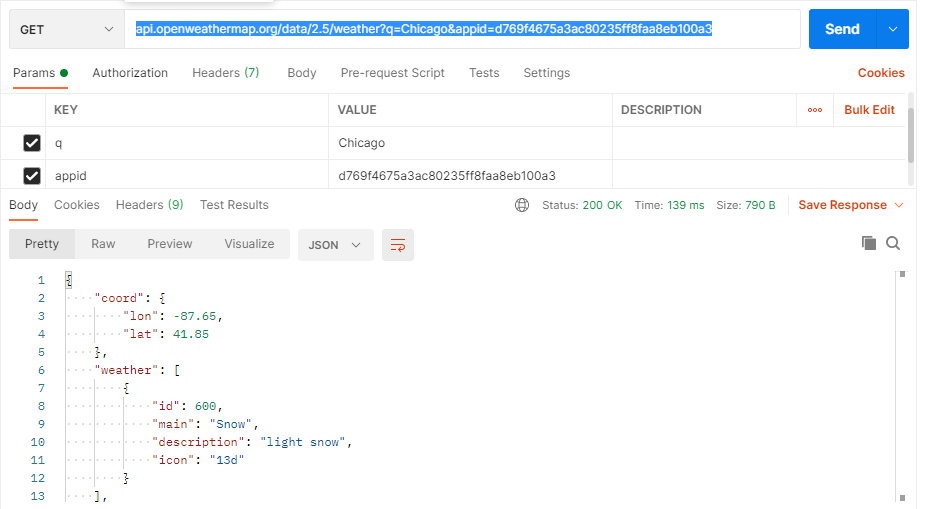
So I already had my Default key which was previously Generated from Openweathermaps. I accidentally also generated a new one called Mobile App Developme.

Afterwards I started testing or weather based on location. First I tried based on City

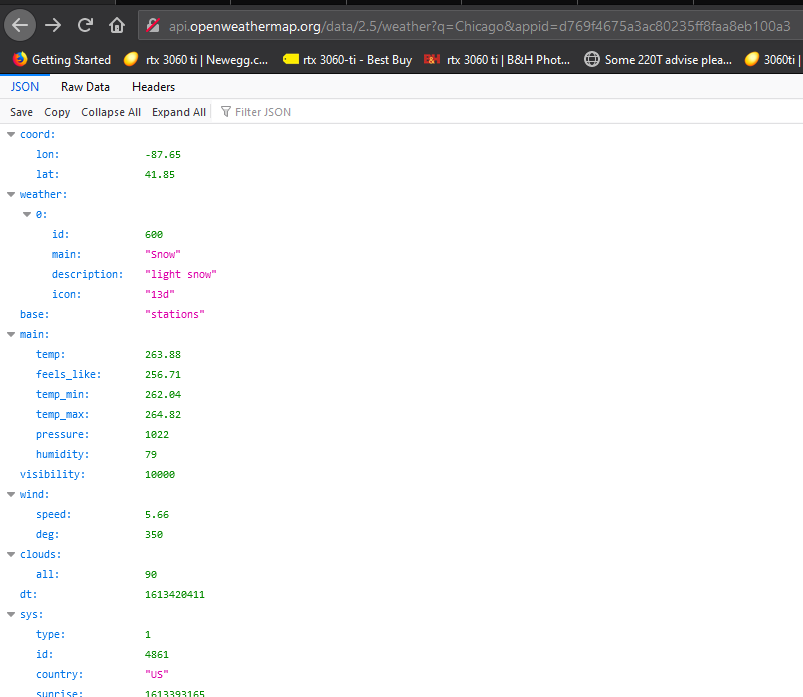
1)



I started with Miami

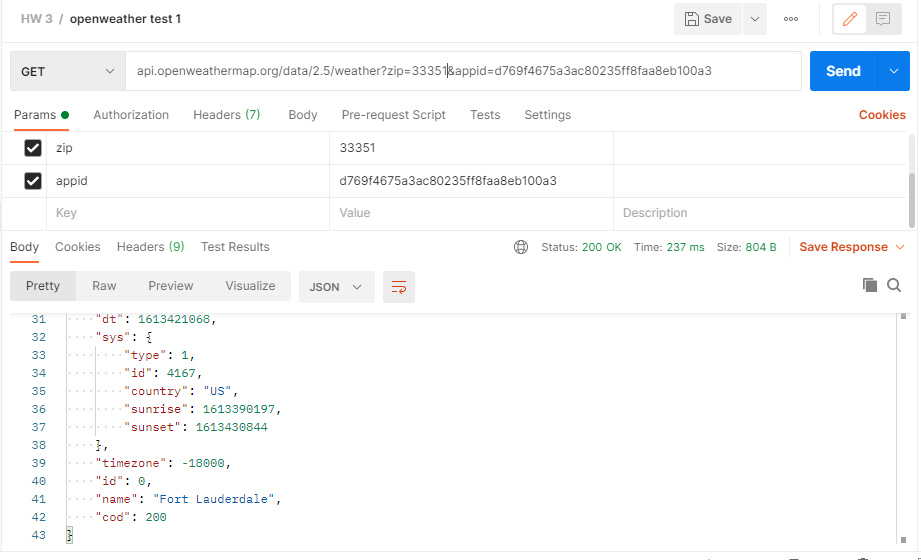


Then Tried Chicago. And it successfully worked as it gave me two separate weather statuses.



I even tried testing it on the web browser after the fact and got the same results so I know it worked based on City.

2)



Next test was Zip Code. The API needed me to specify the country code, but US was set as default so I left it.

I also tested Boa Zip code and Below copied is the Json response.

{

    "coord": {

        "lon": -80.0975,

        "lat": 26.3799

    },

    "weather": [

        {

            "id": 802,

            "main": "Clouds",

            "description": "scattered clouds",

            "icon": "03d"

        }

    ],

    "base": "stations",

    "main": {

        "temp": 301.49,

        "feels\_like": 301.33,

        "temp\_min": 299.82,

        "temp\_max": 302.59,

        "pressure": 1015,

        "humidity": 70

    },

    "visibility": 10000,

    "wind": {

        "speed": 7.2,

        "deg": 170,

        "gust": 12.35

    },

    "clouds": {

        "all": 40

    },

    "dt": 1613422031,

    "sys": {

        "type": 1,

        "id": 3394,

        "country": "US",

        "sunrise": 1613390168,

        "sunset": 1613430789

    },

    "timezone": -18000,

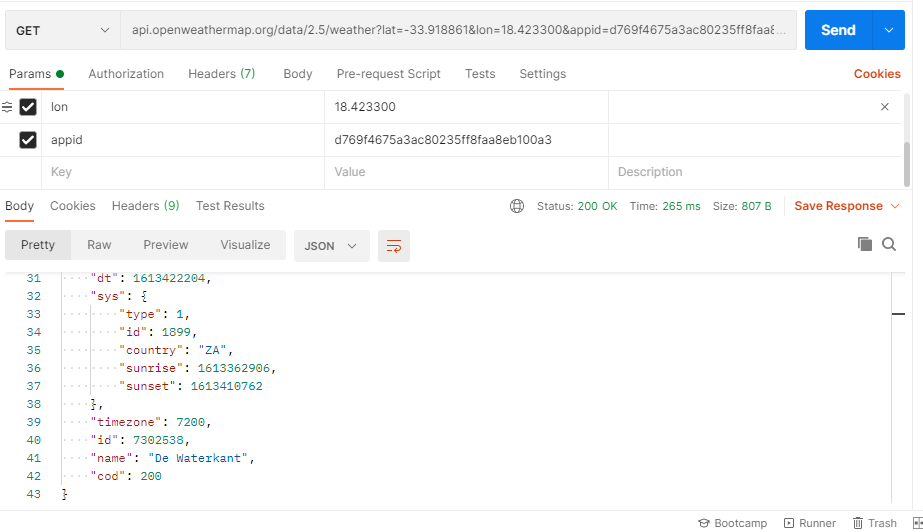
    "id": 0,

    "name": "Boca Raton",

    "cod": 200

}

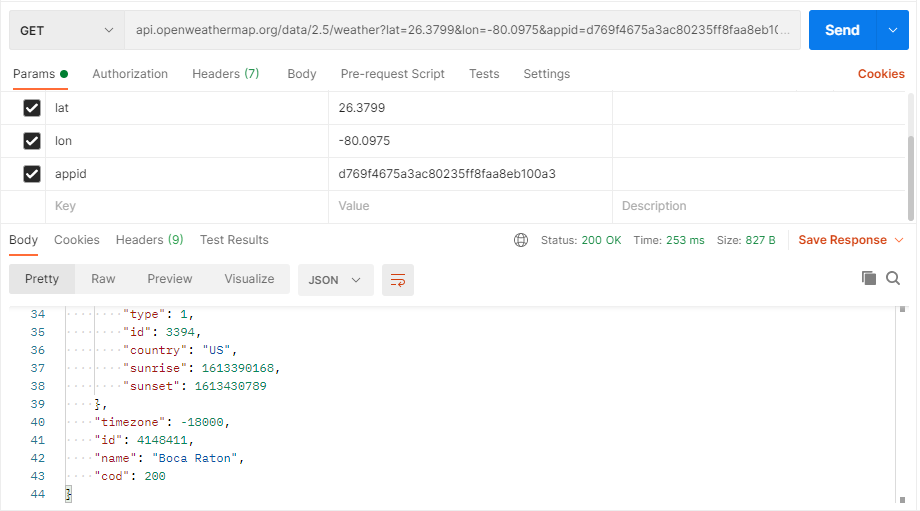
3)



Lastly I got the API for the Geographical location AKA the GPS. Since it demands a longitude and Latitude, as of right now, I have to manually enter it.

In the Slack Chat, one of the fellow students posted this coordinate &lat=-33.918861&lon=18.423300

I tested it out and it worked but I have no clue what location they gave me. So I decide to use the only other location I had on hand.



And Successfully got Boca Raton based on Lon and Lat.