Project 5 – Currency Converter Using Fixer.io (v2)

This is my second attempt at this project. This time I have successfully figured out how to use the API. My process this time was to start from scratch, using only the starter code provided by the professor. Then I signed up for API access with fixer.io, a free API that allows me access to the latest currency rates with a base of EUR. I looked at the documentation and found that if I am to build a currency converter than I must pay for the elevated endpoint access. Instead, I decided to get the latest currency rates with the base rate in Euros (per what the free endpoint allowed), and then funnel the responses into the currency converter function that was supplied by the professor. I then did the conversions myself by setting euros equal to the amount to be converted and divided that by the 'from currency'. Then the converted amount was to be set equal to the 'to currency' multiplied by euros. This allowed me to bypass the limitations of the free API. So, after decoding the JSON object and querying this data, I encoded it into the array named 'data'. After the convert.php file passed the values of the variables along to the functions.php file and got the data retrieved, it set 'converted-currency' equal to the 'currency convert' function. This the values were displayed to the index.php page.

For the debugging process, dumped a lot of 'get' contents into files throughout my program to make sure that the API was returning the proper data. Also, at the beginning of the process I was not even sure I was successfully making calls to the API, so I utilized my API dashboard on fixer.io.

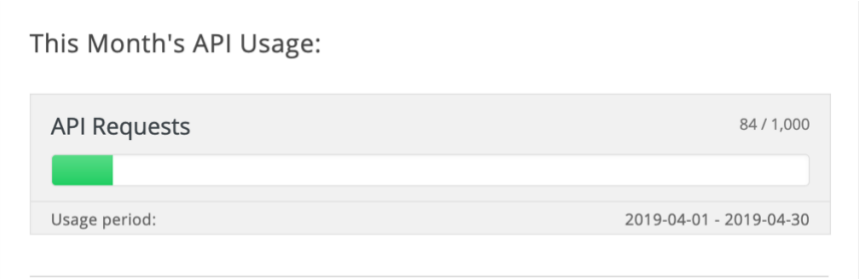

Figure 1 Fixer.io Account Dashboard

I only became satisfied once I saw that the API request count was increasing. Once that was taken care of, the successful API calls, I began to focus on how I could utilize ajax to ensure the user received the information he/she was expecting. There were some errors, however I could not see exactly what the issue was, so I used the JavaScript console that is built into the developer tools. I did this by adding:

```
error: function(XMLHttpRequest, textStatus, errorThrown){
        console.log(textStatus);
        console.log(errorThrown);
```

to my ajax.js file so that I could narrow down the potential problems. I discovered that where I was echoing `json_encode($data);` I should have been returning `json_encode($data);`. This solved the problem and my website was finally functional!

After finally making the conversions and getting the output I wanted, I changed my focus to styling. I kept the same Bootstrap4 template that I have been using since project 3 because I am going with a consistent theme for my portfolio.

All in all, I learned a lot from this exercise. I learned how to integrate a third-party API into my website, how to parse the JSON object and store its contents into variables so that the JSON object can be handled by a php program.