Learning How To Use APIs

One of my favorite learning experiences was using a public API. I used Postman to hit an API's different endpoints. Postman was a great tool to help me visualize the data I was receiving after sending a get request. However, the most important part was knowing what type of requests I could send and what, if any, parameters were needed. Reading the API's documentation was essential in understanding the type of request I could send and what I could expect in return. Once I understood how to access the data I wanted, I was able to start displaying those results in an app. I recently used a public movie API in a small student project. I created a search function using the API's query endpoint and displayed the movie title, year, and director for the text that was entered into the search bar. In this project I console logged the query results so I knew which data points to access and display on the page. Learning about APIs cemented the importance of reading and understanding a software's documentation. I also see the potential of using APIs in future projects and will continue to learn about APIs that require authorization.

Troubleshooting Deployment

For my student project I was having difficulties deploying my app using Heroku. I made sure I had Heroku's required Procfile and that I was using path to send my static files which I had in a public folder. This successfully deployed the html and css file, but I could not get the javascript functionality to work. I started troubleshooting by using the web inspector. The web inspector showed that there were no network errors, which meant my js file was being sent correctly. When I tried to add an item to the list the web inspector logged an error that said it could not access an endpoint on my server file. There was also another error that concerned security, when I tried to send a get request to an API I was using. I fixed this bug by ensuring that the get request had the https protocol. For the access error, I googled the error that Safari and Chrome were displaying. I was able to navigate through a lot of the solutions and errors that others were using and isolate the points that were relevant to my error. This led me to realize that the problem was that the web browsers could not access my localhost server. Since I had deployed my server file to Heroku I had to change my base url from my localhost to the newly created Heroku url. This fixed the error and I was able to use my web app as normal.