Unit Testing Problems

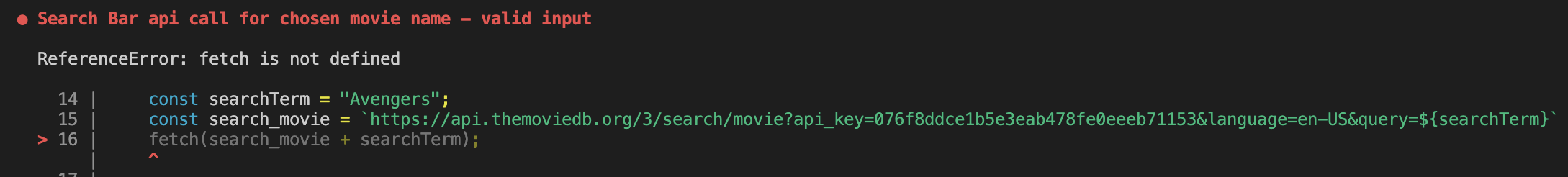
Problems

* Due the nature of our project, all of our scripts manipulate the DOM in some way to properly display what we need from our movie API. By default, jest runs in a node environment so when importing our scripts with DOM syntax, jest did not recognize any of this and therefore threw some errors.
  + Text

    Description automatically generatedFor example when importing our search.js script
* To combat this problem, we had to import an additional jest package called ‘jest-environment-jsdom’ to switch from a node test environment to a DOM environment. With this change we were able to remove most of the problems of DOM syntax, but were still getting tripped up on our event listeners for our button click
  + Text

    Description automatically generatedExample
* After trying to solve this problem, we couldn’t find a better solution then just restructuring our code architecture and putting our DOM elements separate from all of our functions. But for the sake of trying to get the testing working, we commented the event listener part out and attempted to get a test working. So, we tried to create a test for our getMovie function in our search.js script which uses an api call to get a list of movies that the user has specified in the search bar. The problem I ran into was getting jest to recognize ‘getMovie()’ as a function. I made sure to follow proper documentation but I could not get it working
  + Text

    Description automatically generatedGraphical user interface, text

    Description automatically generatedBasic test that checked if the api returned anything at all
* Finally I just decided to mimic the getMovies() function inside my test function to try and get it working. So I ran a fetch(‘api movie search url’) to just see if it worked. The problem I ran into now was that fetch was not recognized because as I have stated, we switched from the node environment to a DOM environment.
* This is when I discovered the jest package ‘jest-fetch-mock’ which mocks a fetch call. This does not actually do a fetch but instead intercepts the fetch call and runs it as it would run from the actual api. Again, I could not get it to work with the actual getMovie() function, but I managed to get it working by recreating the functionality of getMovie()

Text

Description automatically generated

* + Text

    Description automatically generatedFor the test, I just expected the fetch to be returned once
* This was the only test I could get working and I understand that this does not even qualify as a working test because it does not actually implement one of my functions. So if my code changes, it will have no actual effect on the test.