

Remember STAR

Situation: project scenario

Task: your role in project

Action: what you did

Result : what outcome came.

The team made an emulation of spotify's artist page. My role in the project was the related artist page.

Techs used

React

Mongodb: It simplified the database as I didn't have to adhere to strict relational db.

Jest

Enzyme

Express

Styled components: it's great as the styling changes can be bound to each component and styling would be go down the hierarchy which is reminiscent of class inheritance. This provides a clean code structure for me.

For the project it seemed that most of my issues were due to config issues. The first instance of the config issue was when the jest library wouldn't work with the rest of the program. After attempting to debug, I asked about gages configuration and used one that was similar. There was with babel.

Another issue was that enzyme required an adapted which took some time pouring over documentation to get the point.

The other issue was dealing with async and await which ended up abandoning it use for promises for async so I could continue the project.

I chose enzyme over react testing library due to better documentation and that most of the React testing library examples were using await async for the async things. The problem with that was that my configuration wasn't set up well to handle async-await. After attempting to get the configs to play nicely. I made the decision to stick with enzyme as I couldn't spend more time trying to debug configs. This allowed me to continue my project.

The most difficult part was making the choices of the stack and learning the new elements of the stack like jest. Though since jest and enzyme were here a while there was good notes of how to use the testing frameworks.