

IAN FLEMING

Case Study: My 1980s Movies API

Overview:

'My 1980's Movie API' is a web application created using the MERN stack (MongoDB, Express, React, NodeJS). It provides information about 1980s movies (such as plot, director, genre) and allows users to create their own profile, add favourites to their own personal favourites list, and change their personal profile data.

Purpose and context:

The idea behind this project was to demonstrate my skill at building an entire application, from front to back. It was created as part of the CareerFoundry web development bootcamp.

Objective:

My objective was to have a full-stack application built to bolster my portfolio. By building the entire server-side, and bypassing create-react-app for the client-side, the completed app has the potential to impress prospective employers - more than just building a simpler React front-end application.

Approach:

First, I created the back-end by building a REST API using: Node.js, the web-server framework Express.js, and non-relational database MongoDB. The API uses common HTTP methods - GET, POST, PUT, DELETE - which correspond to CRUD (create, read, update, delete) methods - in order to fetch and return information from the database, in the form of

JSON data. For authentication and authorisation, both basic HTTP authentication and JWT authentication were used, depending on context.

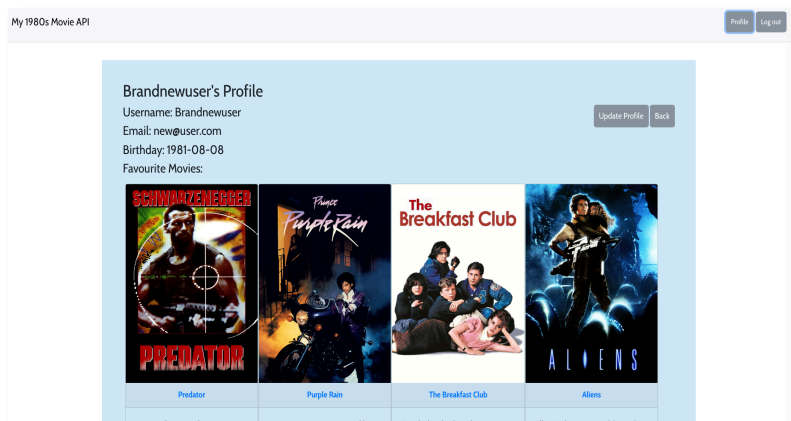
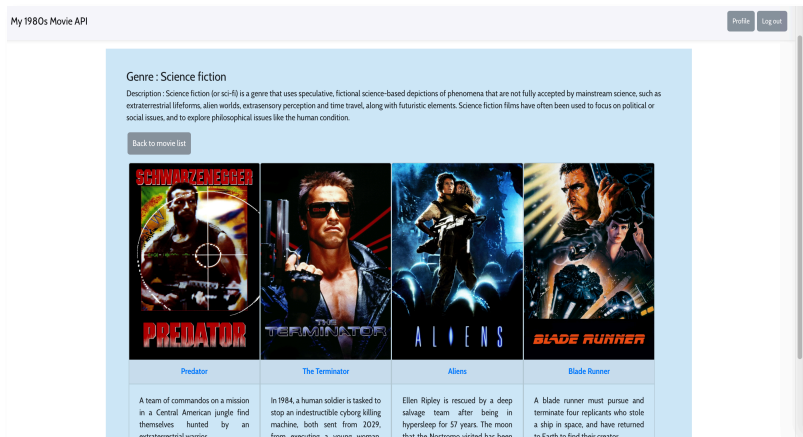
Second, I constructed the client-side. The front-end is an SPA, created with React. Redux is used to ensure state remains consistent throughout and is contained in a single central store. The SPA consists of a variety of views which allow registration, login, displaying all movies, displaying a single movie's information, displaying detailed information regarding a director or a genre, showing the user's saved profile information, adding movies to a personal list of favourites, updating profile information and deleting a user profile. Phew!

Challenges:

The back-end went well. Almost too well! I enjoyed creating it immensely, the specificity and attention-to-detail required appealed to my love of precision. This was where I first began to use the command line too, which was an eye-opener. I became comfortable with it relatively quickly, and it was satisfying to see how much quicker I could perform certain operations via the terminal, as opposed to using the operating system GUI.

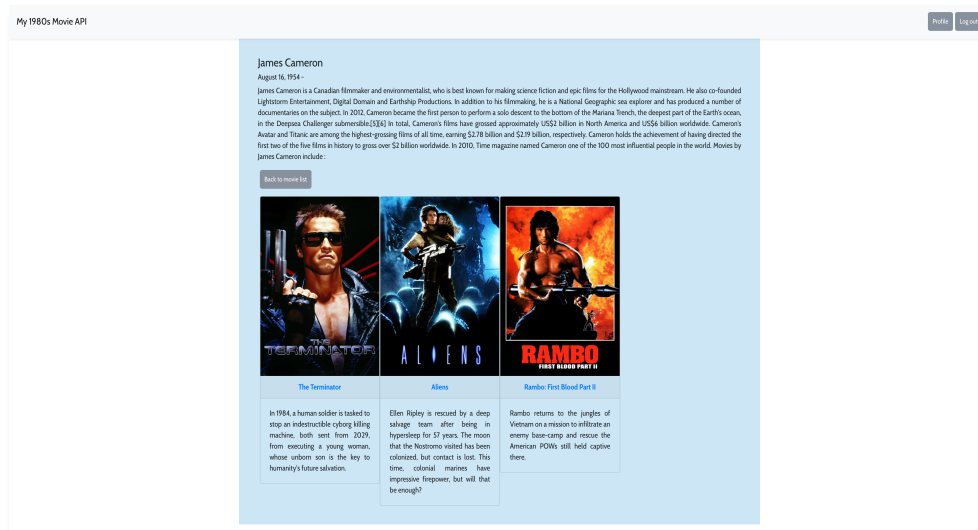
Once I began to construct the front-end, the real challenge began. Having no prior experience with React, it took a while to wrap my head around the concepts. The course teachings require grasping the basics quickly and moving on to more advanced topics. This, combined with a full-time job and two young children, meant it was very difficult to

pour enough time into my learning to ensure full comprehension before implementing what I had learnt. As a result, it felt like I was over-reaching for part of the process - I think that if I had more time to practice and really master the basics, it would have taken me less time to apply the more advanced stuff. Nonetheless, what better way is there to learn than by doing.



I found creating the Profile view, and updating the user info particularly difficult. Redux proved problematic for me too, I felt like my project wasn't sufficiently large to warrant it - but it was good to get some

experience with it. Now, looking back with the benefit of experience and hindsight, I have a more fuller understanding of React as a result of the time spent. Due to the nature of simultaneously working and studying, if I could have had time to learn and perfect the



fundamentals, I think I would have been successful faster. As it stands, it is a project I will return to in the coming months, to both apply my fuller knowledge and improve the codebase, and maybe to build a similar front-end project to show how far I have come. Watch this space! Redux still seems rather confusing, so it's going on my list of 'To learn properly in the future if required'.

Duration:

Up to this point, I was finishing exercises at a rate of a minimum 2 a week. This project soon put a stop to this momentum - two exercises in particular, took a combined total of 6 weeks! All the time spent poring over my screen, scratching my head, staying up way too late, reading and re-reading the documentation and searching Stack Overflow proved useful in the end, as it gave me a better understanding of the pitfalls of the development process, and of React in general.

Three strategies going forward:

- Complete a React course I have purchased, to really consolidate my knowledge of the fundamentals. Simultaneously, build an application from my list of possible React app ideas.
- Take time to brush up on my ES6 knowledge. As well as being incredibly useful for Javascript in 2020, it's absolutely essential if I am going to specialise in React.
- Build the client-side part of app again from scratch. Use the same back-end, but create the front end with a minimum of referral to the course material, using only the advanced

knowledge and experience I've gleaned from the past months. I think this would prove to be an extremely useful exercise.

Credits: Role: Developer - Ian Fleming. Tutor - Roibeard MacUnfraidh. Mentor - Sonia Rose Karungi
