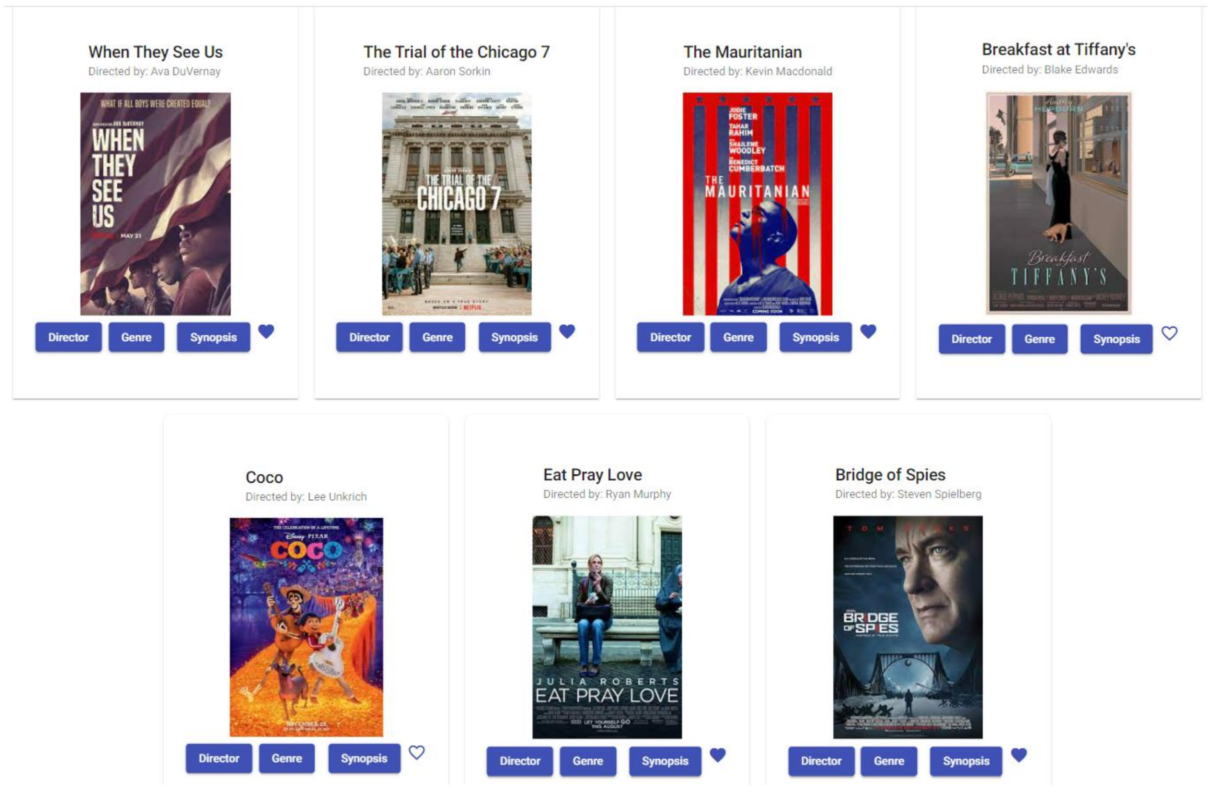
myFlix (Angular) Case Study


When They See Us
Directed by: Ava DuVernay
The Trial of the Chicago 7
Directed by: Aaron Sorkin
The Mauritanian
Directed by: Kevin Macdonald
Breakfast at Tiffany's
Directed by: Blake Edwards
Coco
Directed by: Lee Unkrich
Eat Pray Love
Directed by: Ryan Murphy
Bridge of Spies
Directed by: Steven Spielberg
Director
Genre
Synopsis

Overview

The purpose of this project was to build a complex user interface using Angular JavaScript framework for an application called myFlix based on its existing server-side REST API and MongoDB, and to wire back-end component with the front-end.

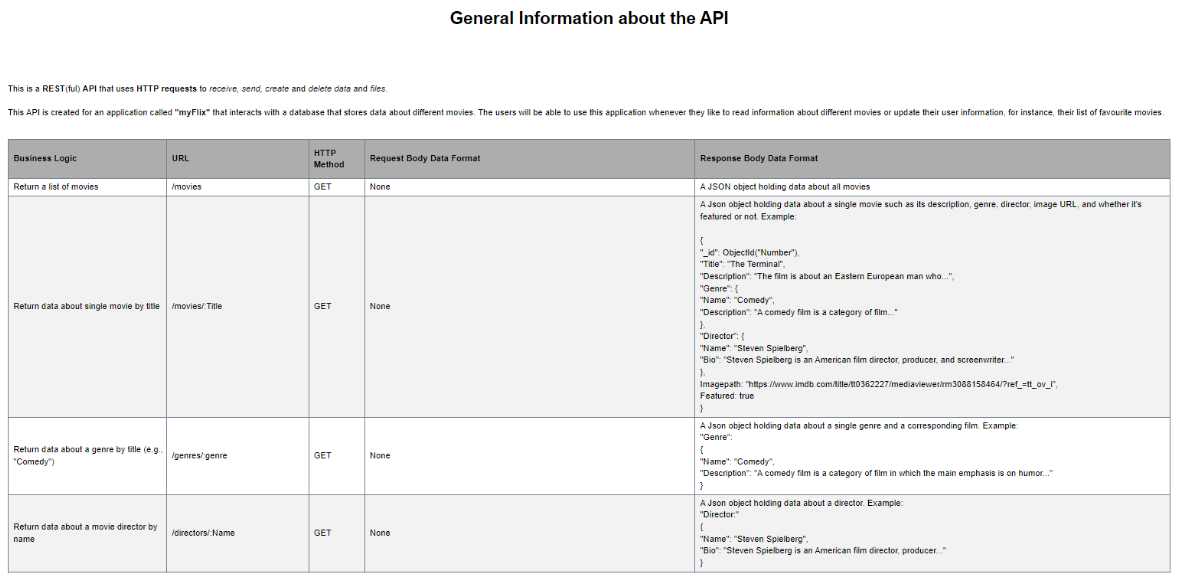
Objective

This project was part of my full-stack web development Career Foundry course. The purpose of it was to acquire some basic knowledge of how Angular framework works, to demonstrate this knowledge by building an interactive user interface and to integrate it with an existing server-side component.

Approach

SERVER-SIDE

This stage involved building a REST API and MongoDB database that would supply information to my client-side about various films, directors, and genres. The server-side was also designed to hold user information such as username, password, date of birth and a list of favourite films. The complete tech stack that I used to accomplish this task was MERN (MongoDB, Express, React and Node.js). As part of this stage of the project, I had to develop a list of API endpoints and write extensive documentation to describe each of them. I also conducted thorough testing of each API endpoint using Postman.



General Information about the API

This is a **REST**(ful) **API** that uses **HTTP requests** to *receive, send, create* and *delete data* and *files*.

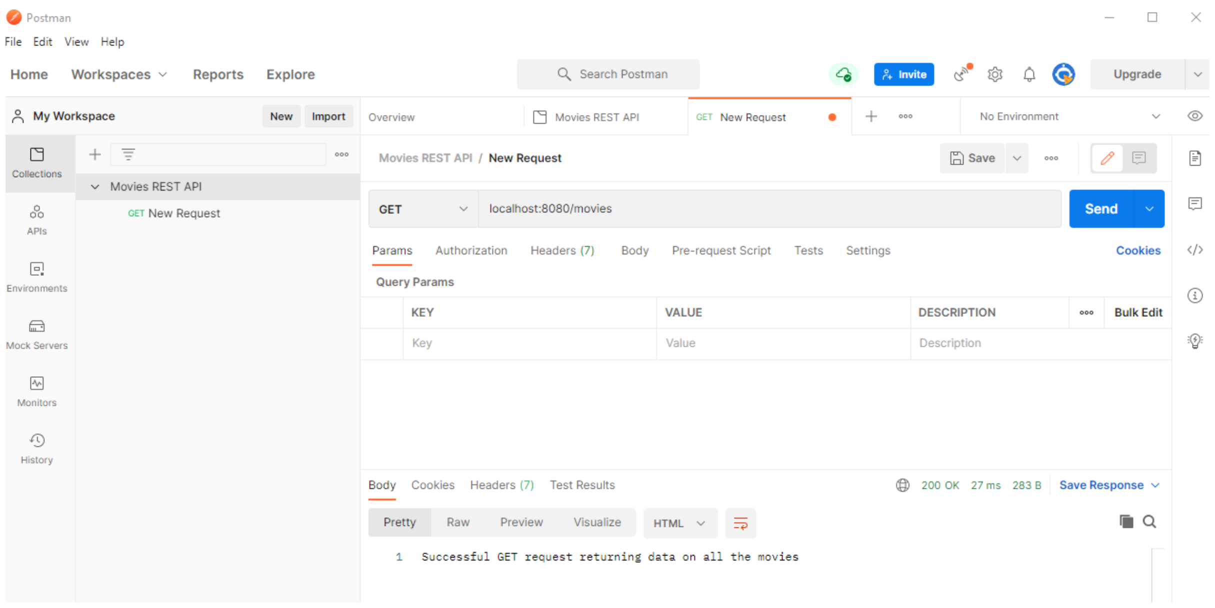
This API is created for an application called **"myFlix"** that interacts with a database that stores data about different movies. The users will be able to use this application whenever they like to read information about different movies or update their user information, for instance, their list of favourite movies.

Business Logic	URL	HTTP Method	Request Body Data Format	Response Body Data Format
Return a list of movies	/movies	GET	None	A JSON object holding data about all movies

Return data about single movie by title	/movies/:Title	GET	None	A Json object holding data about a single movie such as its description, genre, director, image URL, and whether it's featured or not. Example:
				
				{
				"_id": ObjectId("Number"),
				"Title": "The Terminal",
				"Description": "The film is about an Eastern European man who...",
				"Genre": {
				"Name": "Comedy",
				"Description": "A comedy film is a category of film..."
				},
				"Director": {
				"Name": "Steven Spielberg",
				"Bio": "Steven Spielberg is an American film director, producer, and screenwriter..."
				},
				Imagepath: "https://www.imdb.com/title/tt0362227/mediaviewer/rm3088158464/?ref_=tt_ov_i",
				Featured: true
				}

Return data about a genre by title (e.g., "Comedy")	/genres/:genre	GET	None	A Json object holding data about a single genre and a corresponding film. Example:
				"Genre":
				{
				"Name": "Comedy",
				"Description": "A comedy film is a category of film in which the main emphasis is on humor..."
				}

Return data about a movie director by name	/directors/:Name	GET	None	A Json object holding data about a director. Example:
				"Director:"
				{
				"Name": "Steven Spielberg",
				"Bio": "Steven Spielberg is an American film director, producer..."
				}


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
My Workspace New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Overview
No Environment
Save
Send
Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies
Query Params
KEY VALUE DESCRIPTION Bulk Edit
Key Value Description
Body Cookies Headers (7) Test Results
Save Response
Pretty Raw Preview Visualize HTML

Movies REST API
GET New Request

Movies REST API / New Request

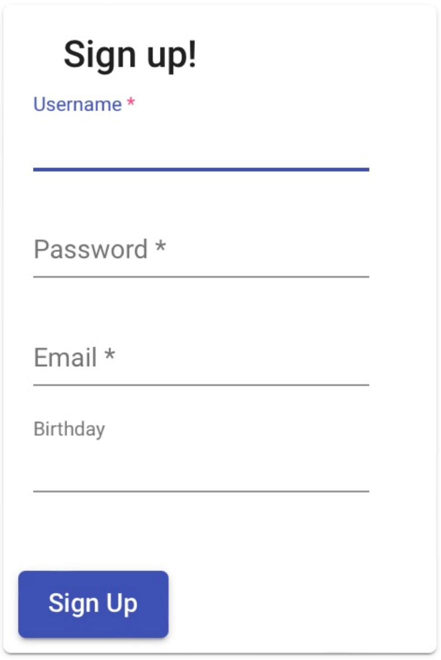
GET localhost:8080/movies

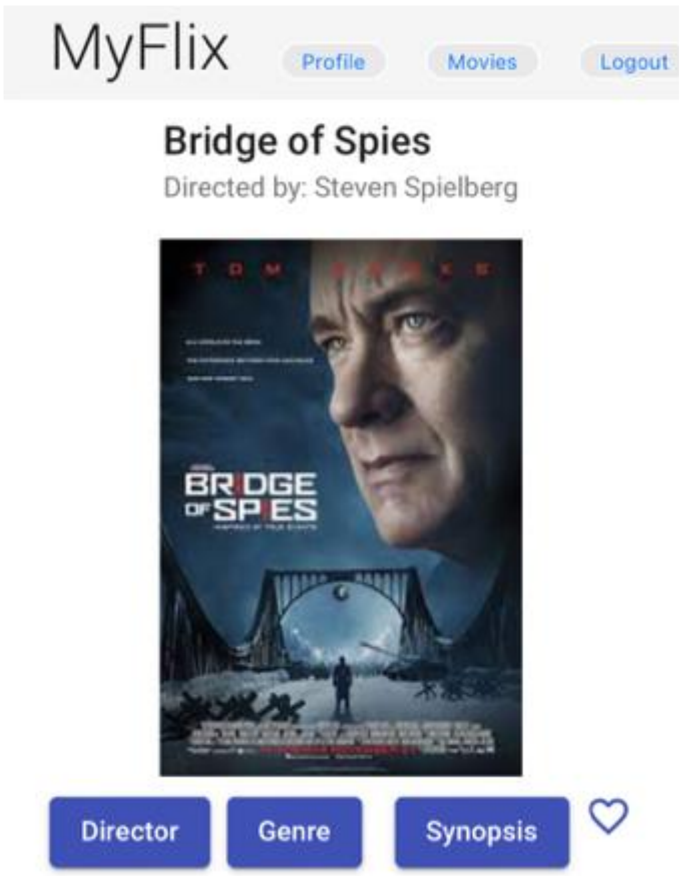
200 OK 27 ms 283 B

```
Successful GET request returning data on all the movies
```

CLIENT-SIDE

At this stage I needed to familiarise myself with the logic and principles of how Angular works. Luckily, I was able to give myself plenty of time to really come to grips with the basics of this framework. I really wanted to gain experience of working with this extremely popular framework that can be used for building a wide range of mobile and desktop applications. Apart from getting to know the materials provided by Career Foundry, I have done a lot of my own research by reading official Angular documentation, researching solutions on stack overflow and by reaching out to my tutor. This made the learning process more manageable and stress free. I came to love using Angular Material, which instantly made the look of this application more consistent and professional. In the end, through quite an enjoyable learning experience, I was able to build an application that included several interface views including registration/login view, main view that displays all films and a user profile view. I believe this approach has helped to tackle quite a steep learning curve that is associated with learning a new platform.





Challenges

Building user interface with Angular was a little challenging because it required me to learn how to work within a new framework. However, after careful research and familiarization with Angular official documentation, I was able to gain enough knowledge to work well within this new platform. It even became my favourite framework because of all the built-in modules and services.

Duration

Learning how to work with Angular to build the client-side of my project took me a little longer than what the estimated time on the course suggested; however, I didn't mind that because I wanted to lay a strong foundation in my knowledge of Angular, so that I can build on this knowledge and acquire more advanced Angular skills in the future.

Conclusions

I am happy with the outcome of this project because I achieved the goal which was to become familiar with the basics of Angular framework, to show the ability to work with it when building user interface and to connect it to the existing server-side component. The published web application demonstrates that I was able to achieve this by building a fully functional website.

When I look back at the process and ask myself what worked well for me, I can immediately name a few things such as taking my time and familiarizing myself with Angular official documentation and trying to wrap my mind around the logic of this new framework, also doing a lot of research on stack overflow to find the solutions, and of course reaching out to my tutor for his advice and guidance.

In regard to what I would do differently, I would probably read all the additional materials that were suggested by the Career Foundry and do the bonus tasks.

In the nearest future, I am planning to familiarise myself with all the additional materials that I haven't read yet, and to do the bonus tasks suggested on the course. I would also like to do an additional Angular course on freecodecamp.org website to advance my Angular skills.