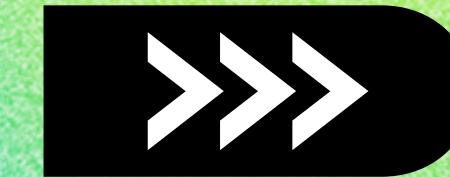


JAMIE TRACY

MOVIE API

A FULL STACK CASE STUDY

OVERVIEW



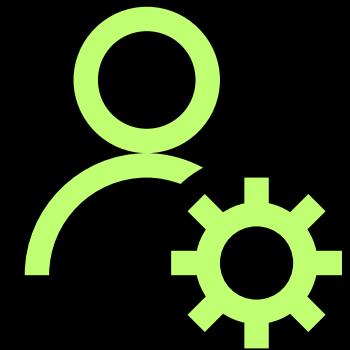
This REST API for the CineDEX web application interacts with a database that stores data about different movies.

[See my code here](#)

OBJECTIVE

This project was part of the full-stack immersion course from the web development program at CareerFoundry to build the server-side component of a “movies” web application to showcase in my portfolio. The web application will provide users with access to information about different movies, directors, and genres. Users will be able to sign up, update their personal information, and create a list of their favorite movies.

DETAILS



Roles

Role: Lead Developer
Tutor: Ebere Iweala
Mentor: Mahesh Rodrigo



Duration

This project took me approximately 3 weeks to complete from start to finish.



Methodologies

- MongoDB
- Express
- Node.js
- REST API



THE PROCESS

GETTING SET UP

I created a REST API to interact with a non-relational database of movies I made using MongoDB. The API is accessed through HTTP methods, where requests can be sent to set endpoints to retrieve specific data from the database.

AUTHENTICATION

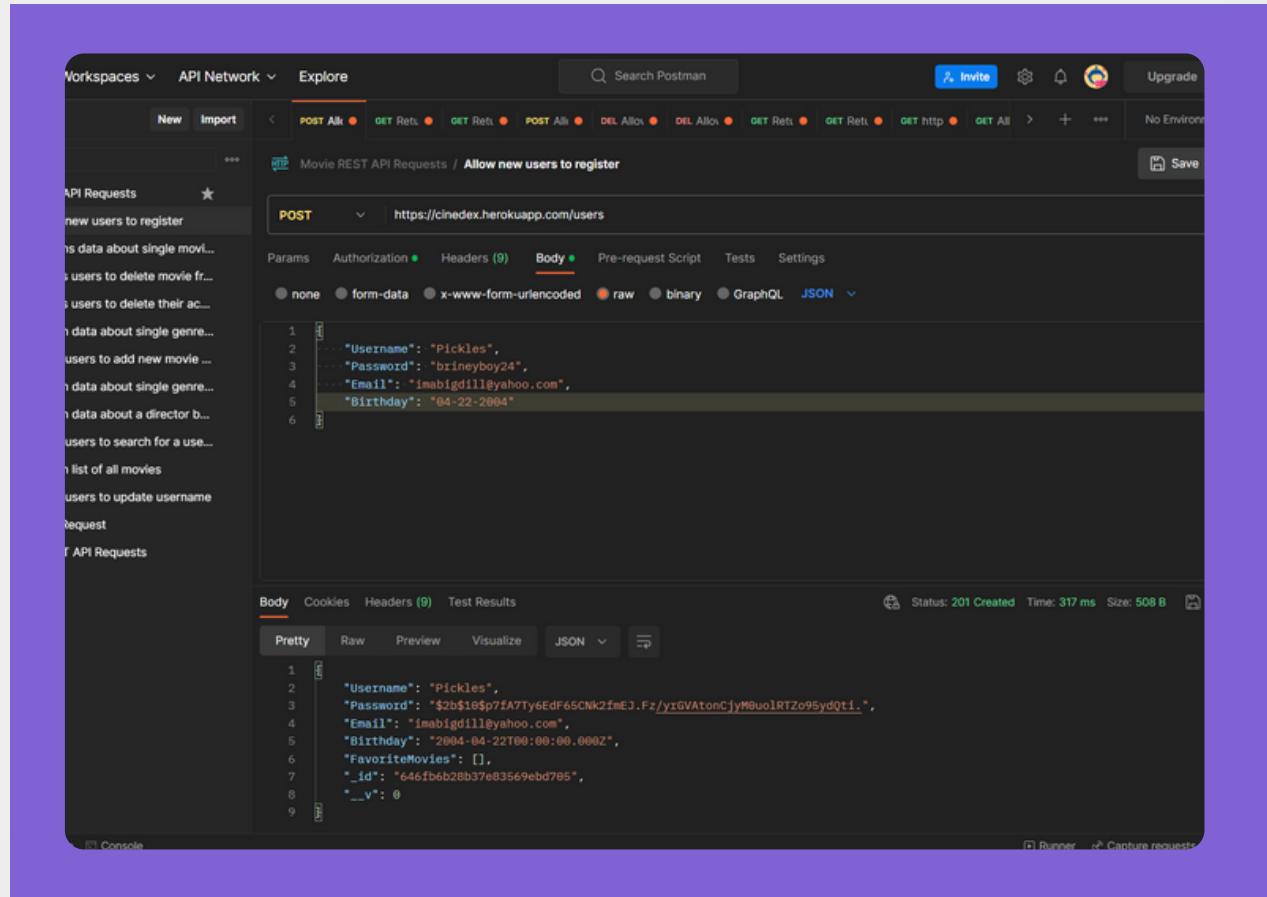
I also implemented basic HTTP and JWT token-based authentication as well as password hashing and data validation.

TESTING

I was able to test the functionality of these endpoints using Postman during development.

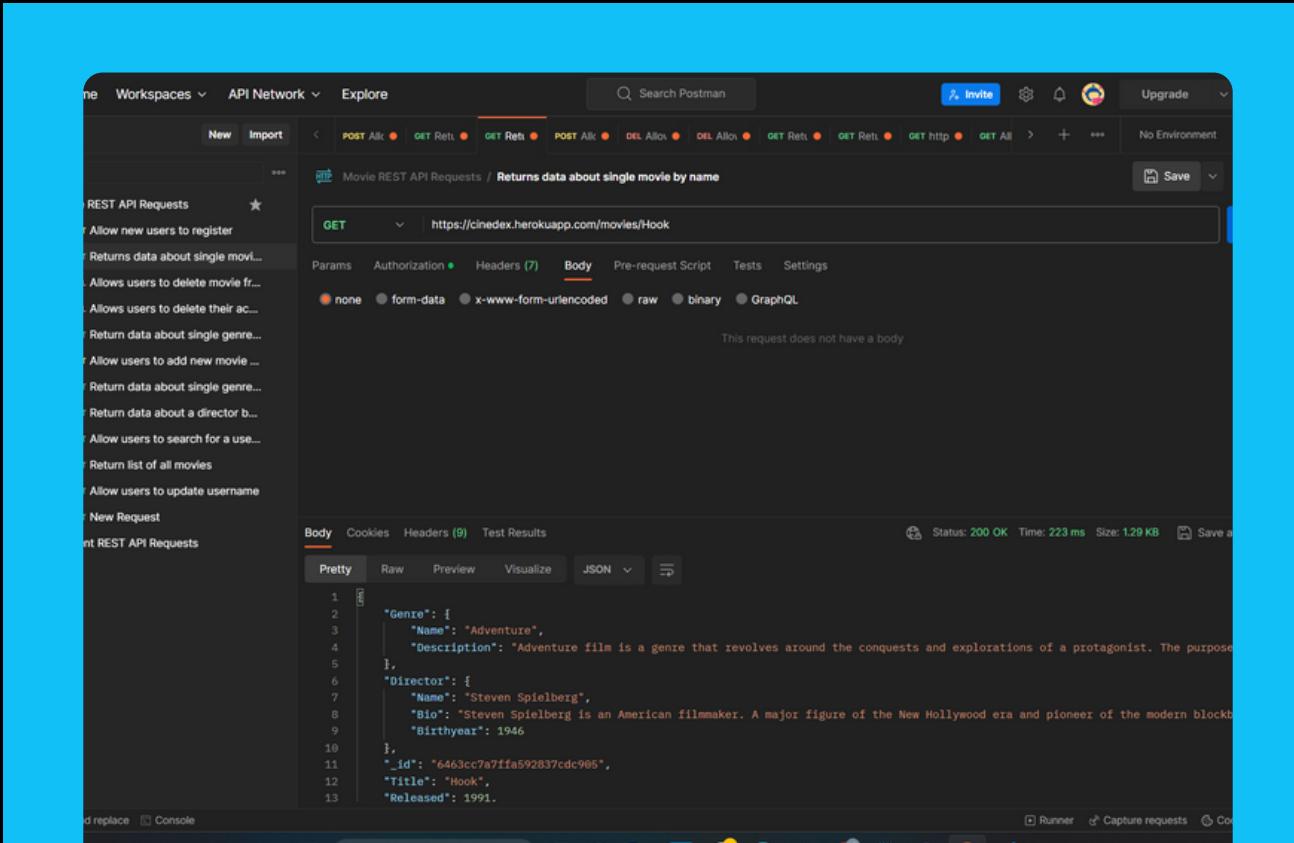
DEPLOYMENT

Last, I used MongoDB Atlas to host my database and Heroku to deploy my app.



SCREENSHOTS

TESTING ENDPOINTS IN POSTMAN



DATABASE ENTRIES

```
CinedexDB> db.users.deleteOne({ Username: "Bill" })
{
  acknowledged: true,
  deletedCount: 1
}
CinedexDB> db.users.find()
[
  {
    _id: ObjectId("6463d50f7ffa592837cdc91d"),
    Username: 'Maria',
    Password: 'eey0bre',
    Email: 'westsidestory@gmail.com',
    Birthday: ISODate("1960-07-19T00:00:00.000Z"),
    FavoriteMovies: [ ObjectId("6463cc7a7ffa592837cdc905") ]
  },
  {
    _id: ObjectId("6463e1817ffa592837cdc91e"),
    Username: 'Spongebob',
    Password: 'Gary549',
    Email: 'pineappleunderthesea@msn.com',
    Birthday: ISODate("1990-05-25T00:00:00.000Z"),
    FavoriteMovies: [ ObjectId("6463cc7a7ffa592837cdc90b") ]
  },
  {
    _id: ObjectId("6463e2a27ffa592837cdc91f"),
    Username: 'Luigi',
    Password: 'itsameligigi2',
    Email: 'hauntedmansion3@hotmail.com',
    Birthday: ISODate("1958-03-13T00:00:00.000Z"),
    FavoriteMovies: [
      ObjectId("6463cc7a7ffa592837cdc906"),
      ObjectId("6463cc7a7ffa592837cdc905")
    ]
  },
  {
    _id: ObjectId("6463e36c7ffa592837cdc920"),
    Username: 'Princess Peach',
    Password: 'one2bucklemyshoe',
    Email: 'princesspeach@aol.com',
    Birthday: ISODate("1970-12-23T00:00:00.000Z"),
    FavoriteMovies: [
      ObjectId("6463cc7a7ffa592837cdc90b"),
      ObjectId("6463cc7a7ffa592837cdc905")
    ]
  }
]
```

RETROSPECTIVE

WHAT WENT WELL	CHALLENGES	FINAL THOUGHTS
As a very organized person and analytical thinker, I really enjoyed working on the database and felt I was able to complete this project fairly quickly and efficiently.	I had some issues at first implementing the user authentication and authorization portion and password hashing. After some fine tuning and reviewing the HTTP requests and responses using Postman I was able to catch my mistakes and fix them.	While I found myself getting frustrated trying to figure out some of the error messages I was receiving in Postman, I learned a lot about recognizing my errors and finding resolutions to them independently in this project. This skill has helped me tremendously in subsequent projects as well.