Oscar Movie Data Microservices Overview

Objective: To develop robust microservices for efficient management and access to Oscar movie data, enhancing user experience and data accessibility.

Technologies Used:

- **Maven**: Simplifies project build and management
- **Spring Boot**: Facilitates rapid development and deployment
- **Lombok**: Reduces boilerplate code, improving readability
- **MySQL**: Offers reliable database storage solutions
- **Eureka Server**: Ensures seamless service discovery and integration

Oscar Movie Data Microservices Overview

Microservices:

- **Genre Service**: Provides comprehensive genre information, enabling dynamic content filtering and search functionality.
- **Movie Service**: Delivers detailed Oscar movie data including ID, name, cast, and genre, with a focus on enriching the user's exploration experience.

Key Features:

- Secure API endpoints to safeguard data integrity and privacy.
- Swagger documentation for clear API guidelines and testing.
- Exception handling and logging for robust error management and diagnostics.
- Eureka Server registration for improved microservice orchestration and scalability.

Microservices Architecture

Genre Service

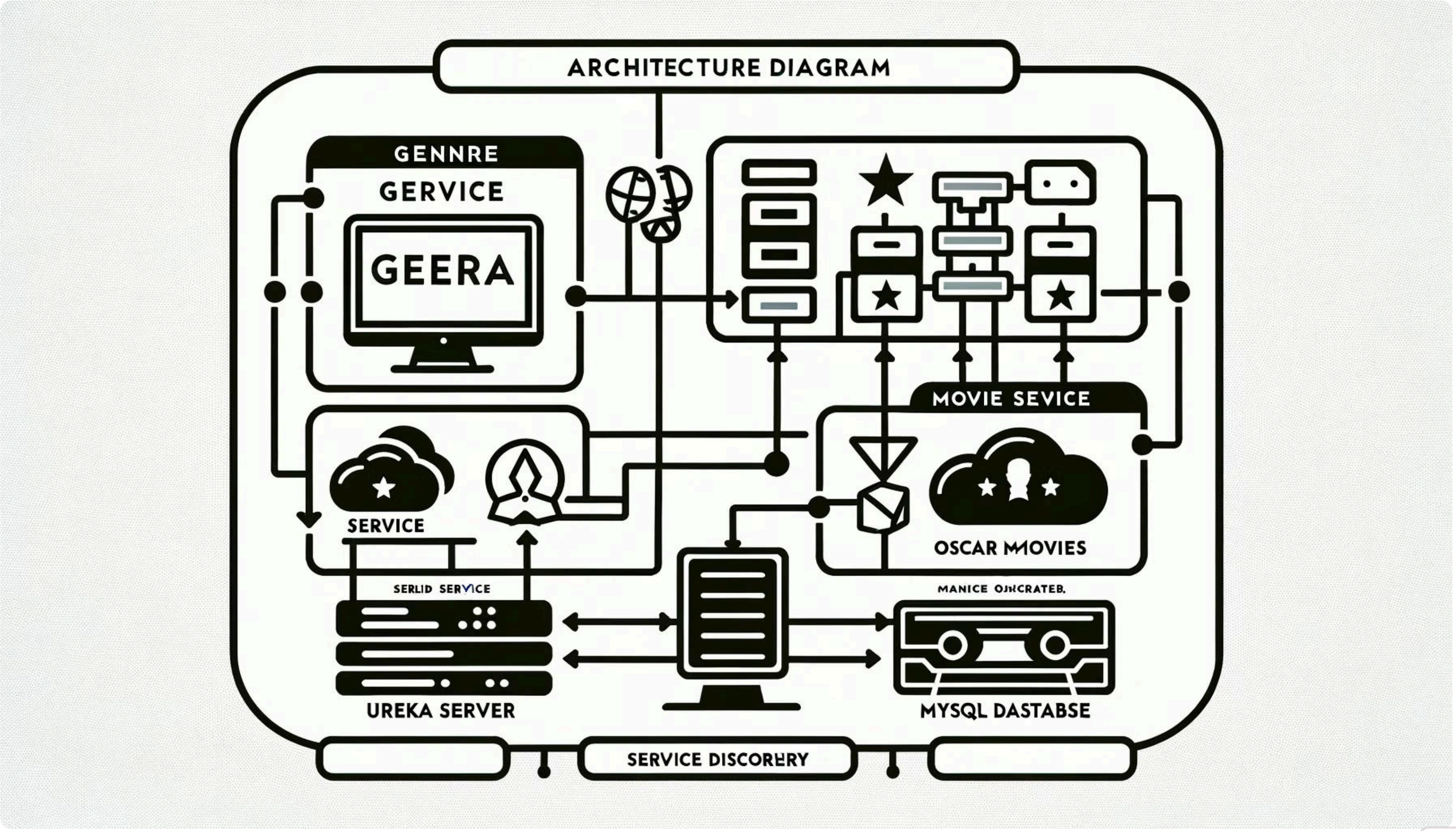
This service is responsible for providing comprehensive genre information. It interacts with the MySQL Database to retrieve genre data and makes this information available to other services or clients.

Movie Service

Delivers detailed data about Oscar movies. It queries both its own information from the MySQL Database and integrates genre information by communicating with the Genre Service.

Eureka Server and MySQL Database

Facilitators of microservices orchestration, service discovery, and efficient data storage, forming the backbone of the architectural framework.



Genre Service

Specialized in managing and providing access to genre information.

API Endpoints:

1. **Retrieve Genre Details**
   - **Endpoint**: GET /genres/{genreId}
   - **Description**: Fetches details for a specific genre by its ID.
2. **List All Genres**
   - **Endpoint**: GET /genres
   - **Description**: Returns a list of all available genres in the system.
3. **Add a New Genre**
   - **Endpoint**: POST /genres
   - **Description**: Allows administrators to add a new genre to the database.
4. **Update Genre Information**
   - **Endpoint**: PUT /genres/{genreId}
   - **Description**: Enables administrators to update existing genre information by ID.

Movie Service

The Movie Service stores Oscar movie info and works with the Genre Service to add genre details.

API Endpoints:

1. **Retrieve Movie Details**
   - **Endpoint**: `GET /movies/{movieId}`
   - **Functionality**: Fetches complete details of a movie identified by its ID, including its name, cast, and associated genre. It ensures users have access to rich and detailed movie descriptions.
2. **List All Movies**
   - **Endpoint**: `GET /movies`
   - **Functionality**: Provides a catalog of all available movies in the database, enabling users to browse the collection and select movies of interest.
3. **Add a New Movie**
   - **Endpoint**: `POST /movies`
   - **Access Level**: Admin Only
   - **Functionality**: Allows administrators to introduce new movies to the collection, ensuring the database is up-to-date with the latest Oscar contenders and winners.
4. **Update Movie Information**
   - **Endpoint**: `PUT /movies/{movieId}`
   - **Access Level**: Admin Only
   - **Functionality**: Grants administrators the ability to update existing movie records, such as correcting details or adding new information, to maintain the accuracy and relevance of the movie database.

Database Schema Design

Embracing a clearly delineated visual schema that encapsulates the tables and their intricate relationships, the Database Schema optimally captures the essence of both the Movies and Genres database components.

Genres Table

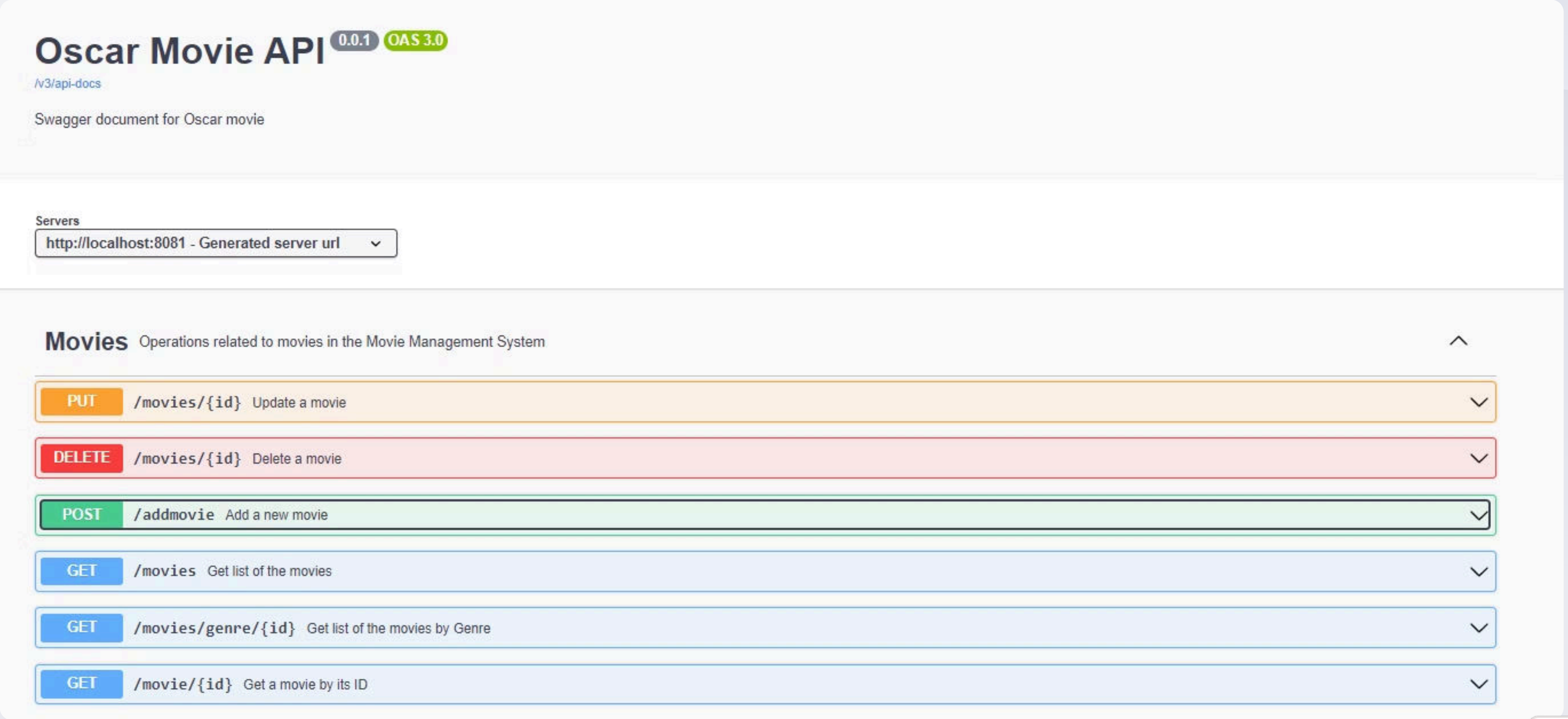
Column Name	Data Type	Constraints	Description
genre_id	INT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for the genre
name	VARCHAR(255)	NOT NULL, UNIQUE	Name of the genre
description	TEXT		Description of the genre

Movies Table

Column Name	Data Type	Constraints	Description
movie_id	INT	AUTO_INCREMENT, PRIMARY KEY	Unique identifier for the movie
name	VARCHAR(255)	NOT NULL	Name of the movie
cast	TEXT	NOT NULL	Cast involved in the movie
genre_id	INT	FOREIGN KEY(genre_id) REFERENCES genres(genre_id)	ID of the genre this movie belongs to
movie_description	TEXT	NOT NULL	Description of the movie

API Documentation with Swagger

Experience the numerous benefits of leveraging Swagger for API testing and documentation through a dynamic and intuitive showcase of the API's endpoints within the Swagger UI. This interface offers unparalleled visibility into documentation, testing, and endpoint navigation.



Oscar Movie API 0.0.1 OAS 3.0

/v3/api-docs

Swagger document for Oscar movie

Servers

http://localhost:8081 - Generated server url

Movies Operations related to movies in the Movie Management System

PUT `/movies/{id}` Update a movie

DELETE `/movies/{id}` Delete a movie

POST `/addmovie` Add a new movie

GET `/movies` Get list of the movies

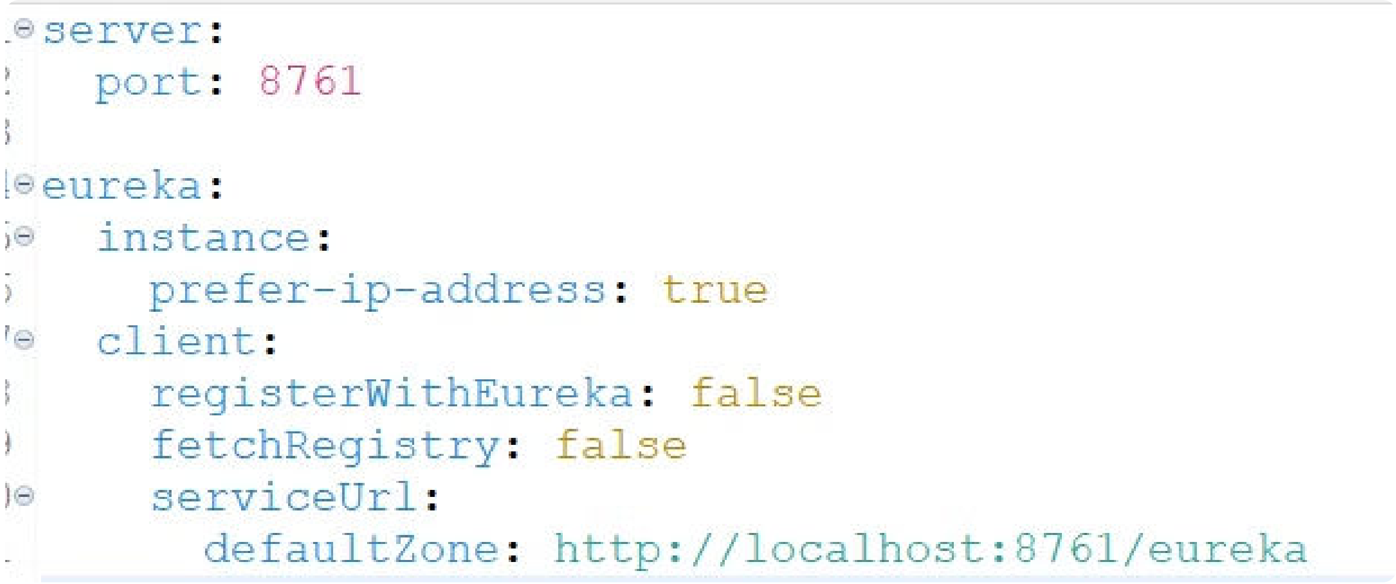
GET `/movies/genre/{id}` Get list of the movies by Genre

GET `/movie/{id}` Get a movie by its ID

Eureka Server Configuration

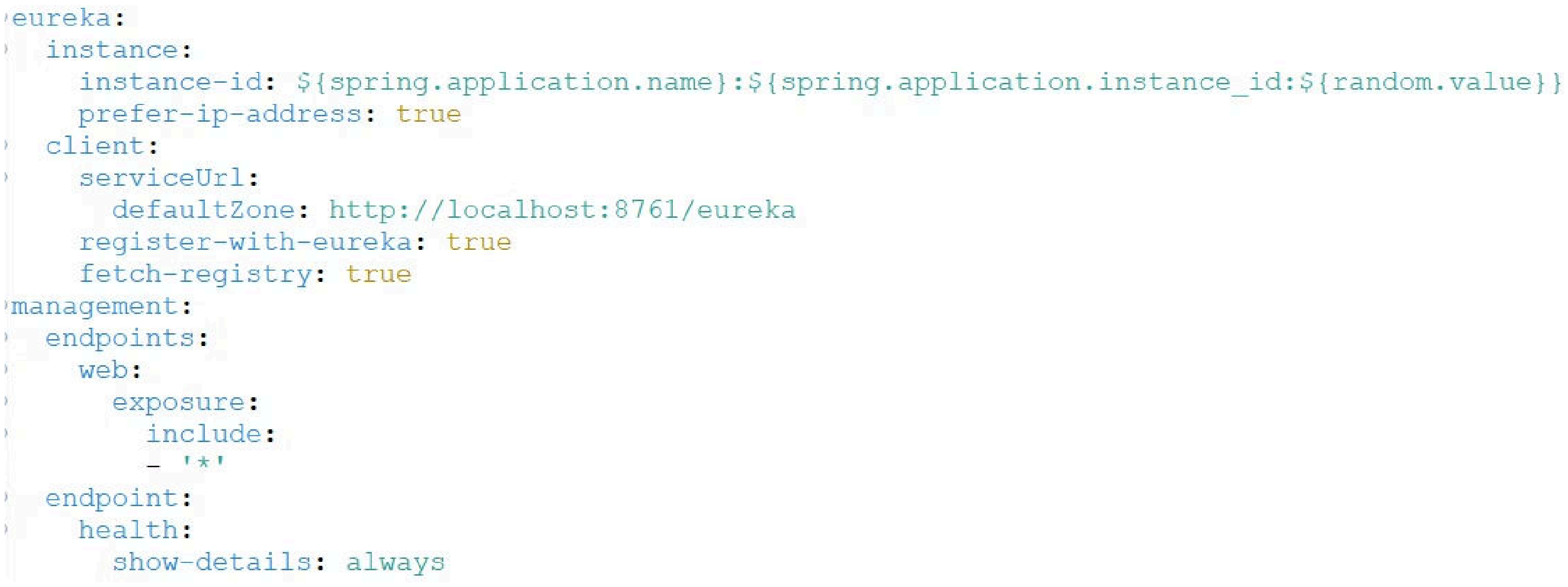
Explore the pivotal role of the Eureka Server within the microservices architecture, underpinning robust service registration and seamless discovery mechanisms. The configuration encompasses highlights of service registration and discovery within the dynamic Eureka environment.

Server:



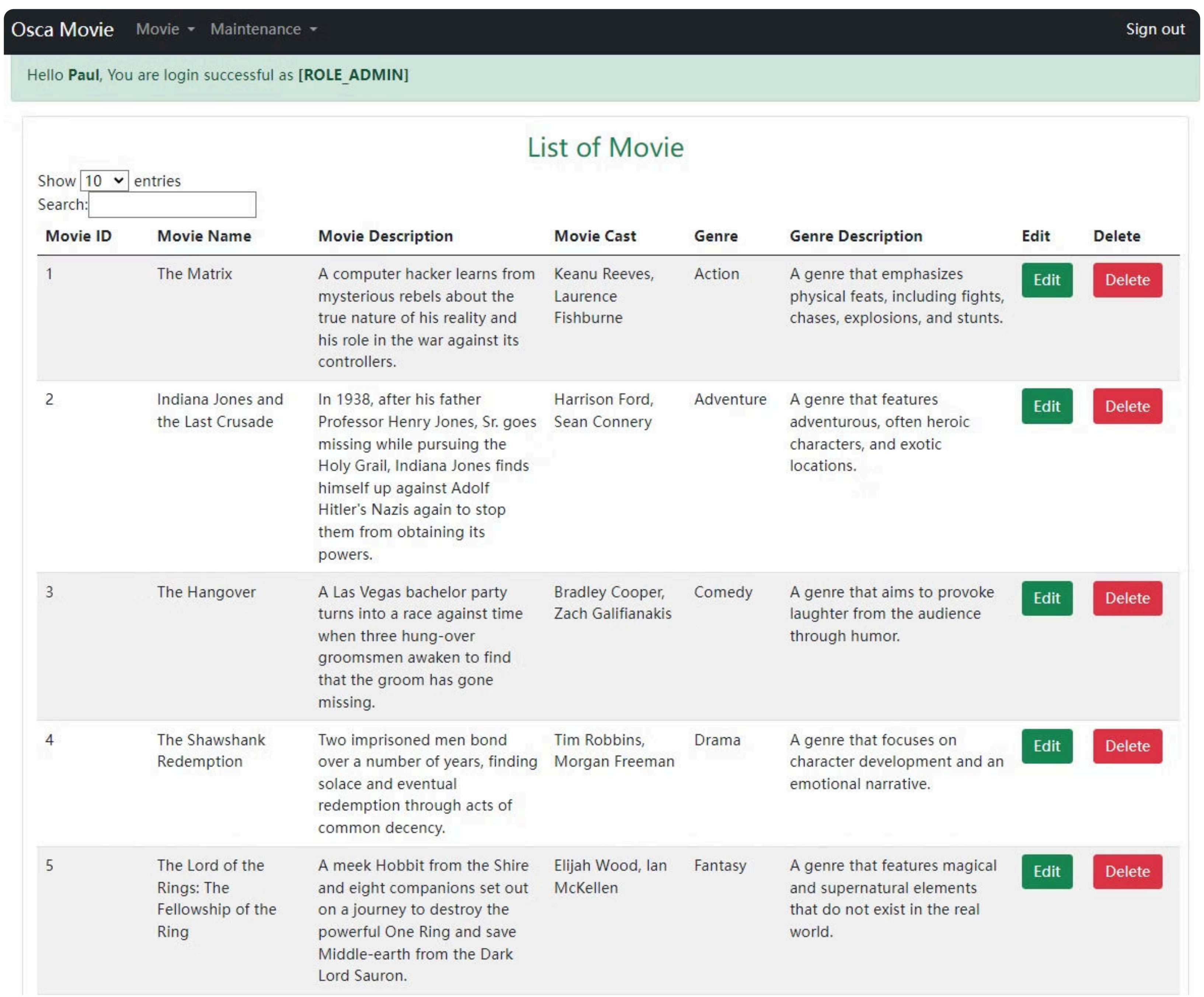
```yaml
server:
  port: 8761

eureka:
  instance:
    prefer-ip-address: true
  client:
    registerWithEureka: false
    fetchRegistry: false
    serviceUrl:
      defaultZone: http://localhost:8761/eureka
```

Client:



```yaml
eureka:
  instance:
    instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
    prefer-ip-address: true
  client:
    serviceUrl:
      defaultZone: http://localhost:8761/eureka
    register-with-eureka: true
    fetch-registry: true
management:
  endpoints:
    web:
      exposure:
        include:
        - '*'
  endpoint:
    health:
      show-details: always
```

Web UI

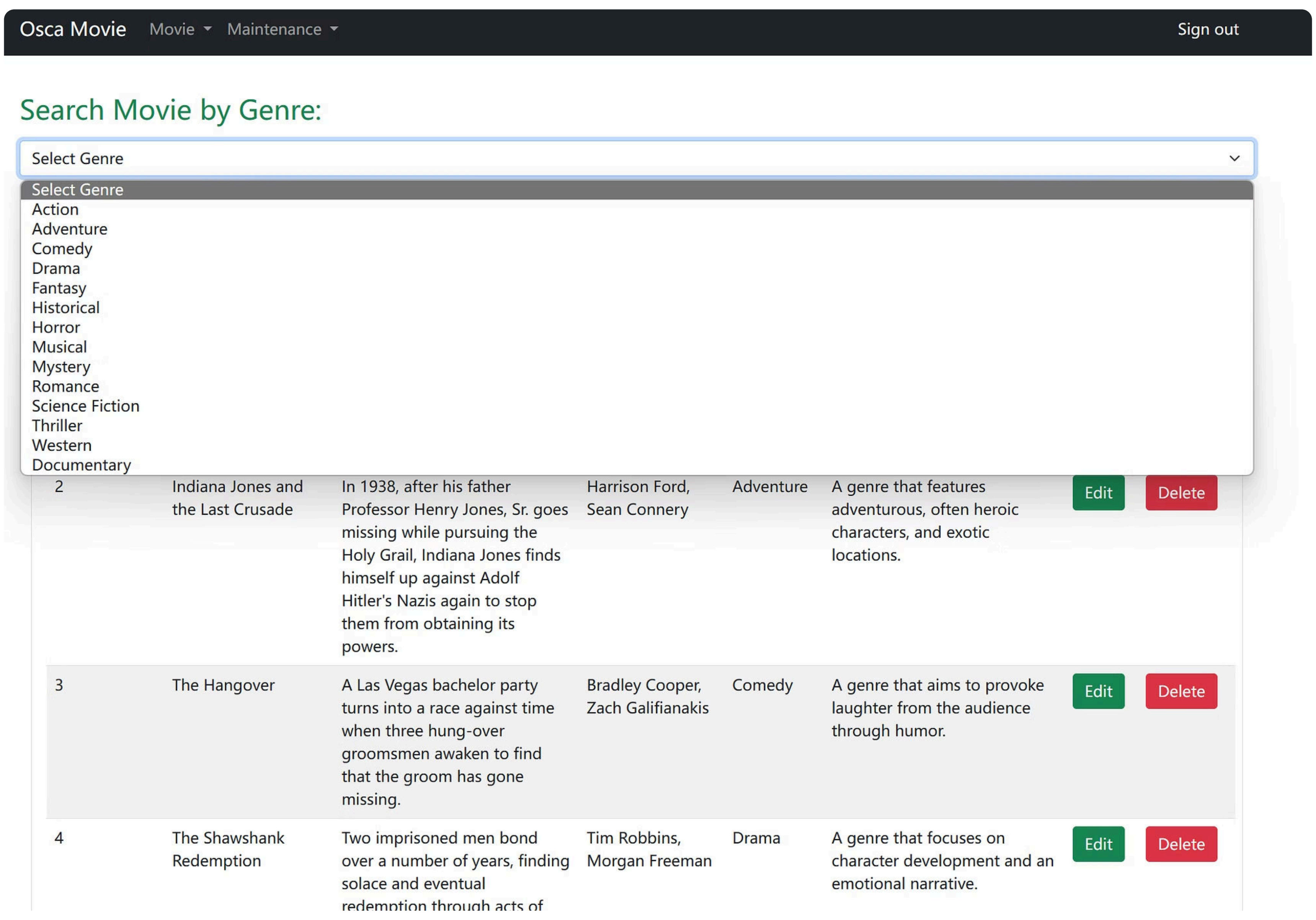


Hello **Paul**, You are login successful as **[ROLE_ADMIN]**

List of Movie

Show 10 entries
Search:

Movie ID	Movie Name	Movie Description	Movie Cast	Genre	Genre Description	Edit	Delete
1	The Matrix	A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.	Keanu Reeves, Laurence Fishburne	Action	A genre that emphasizes physical feats, including fights, chases, explosions, and stunts.	Edit	Delete
2	Indiana Jones and the Last Crusade	In 1938, after his father Professor Henry Jones, Sr. goes missing while pursuing the Holy Grail, Indiana Jones finds himself up against Adolf Hitler's Nazis again to stop them from obtaining its powers.	Harrison Ford, Sean Connery	Adventure	A genre that features adventurous, often heroic characters, and exotic locations.	Edit	Delete
3	The Hangover	A Las Vegas bachelor party turns into a race against time when three hung-over groomsmen awaken to find that the groom has gone missing.	Bradley Cooper, Zach Galifianakis	Comedy	A genre that aims to provoke laughter from the audience through humor.	Edit	Delete
4	The Shawshank Redemption	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.	Tim Robbins, Morgan Freeman	Drama	A genre that focuses on character development and an emotional narrative.	Edit	Delete
5	The Lord of the Rings: The Fellowship of the Ring	A meek Hobbit from the Shire and eight companions set out on a journey to destroy the powerful One Ring and save Middle-earth from the Dark Lord Sauron.	Elijah Wood, Ian McKellen	Fantasy	A genre that features magical and supernatural elements that do not exist in the real world.	Edit	Delete

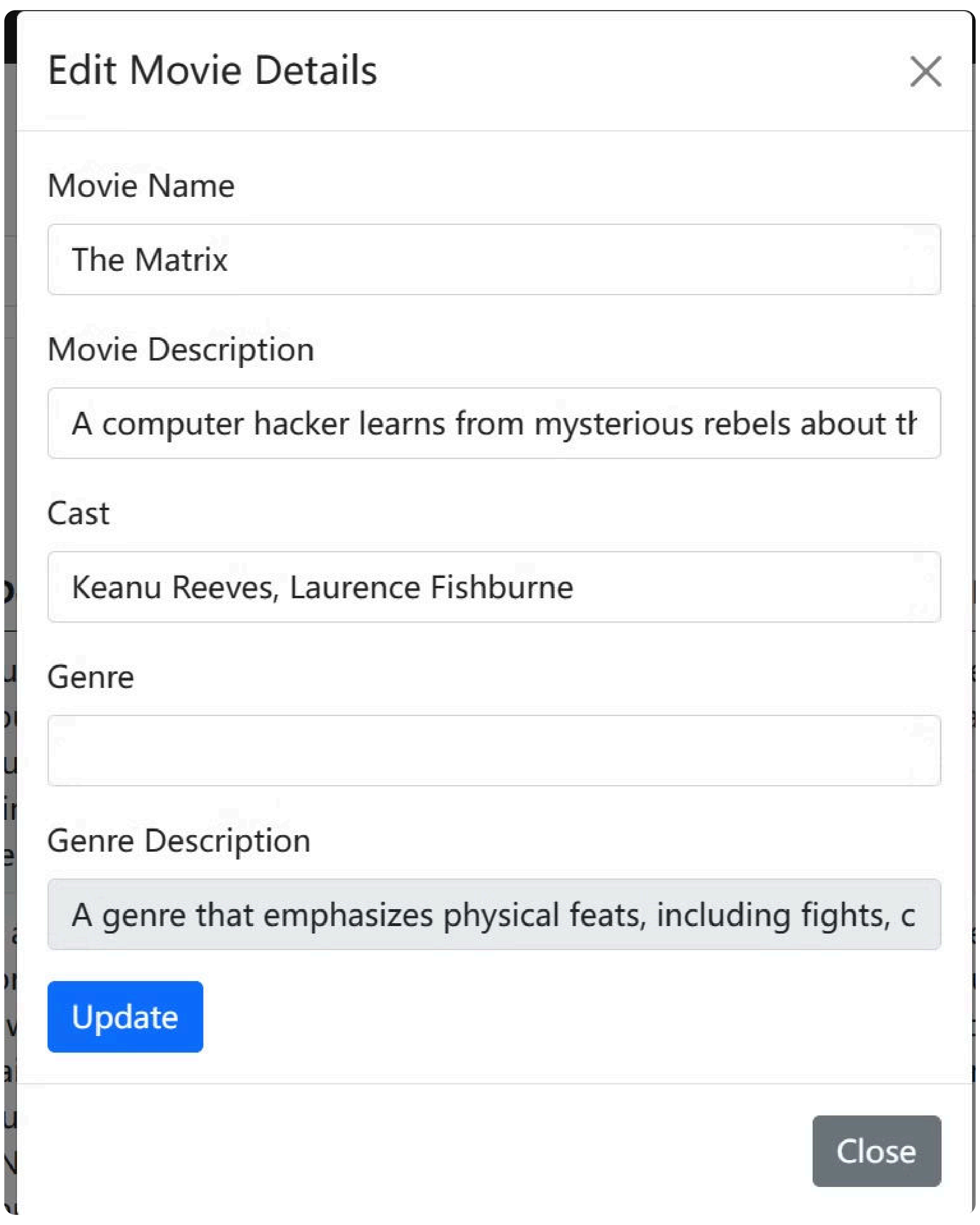


Search Movie by Genre:

Select Genre

- Select Genre
- Action
- Adventure
- Comedy
- Drama
- Fantasy
- Historical
- Horror
- Musical
- Mystery
- Romance
- Science Fiction
- Thriller
- Western
- Documentary

2	Indiana Jones and the Last Crusade	In 1938, after his father Professor Henry Jones, Sr. goes missing while pursuing the Holy Grail, Indiana Jones finds himself up against Adolf Hitler's Nazis again to stop them from obtaining its powers.	Harrison Ford, Sean Connery	Adventure	A genre that features adventurous, often heroic characters, and exotic locations.	Edit	Delete
3	The Hangover	A Las Vegas bachelor party turns into a race against time when three hung-over groomsmen awaken to find that the groom has gone missing.	Bradley Cooper, Zach Galifianakis	Comedy	A genre that aims to provoke laughter from the audience through humor.	Edit	Delete
4	The Shawshank Redemption	Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of	Tim Robbins, Morgan Freeman	Drama	A genre that focuses on character development and an emotional narrative.	Edit	Delete



Edit Movie Details

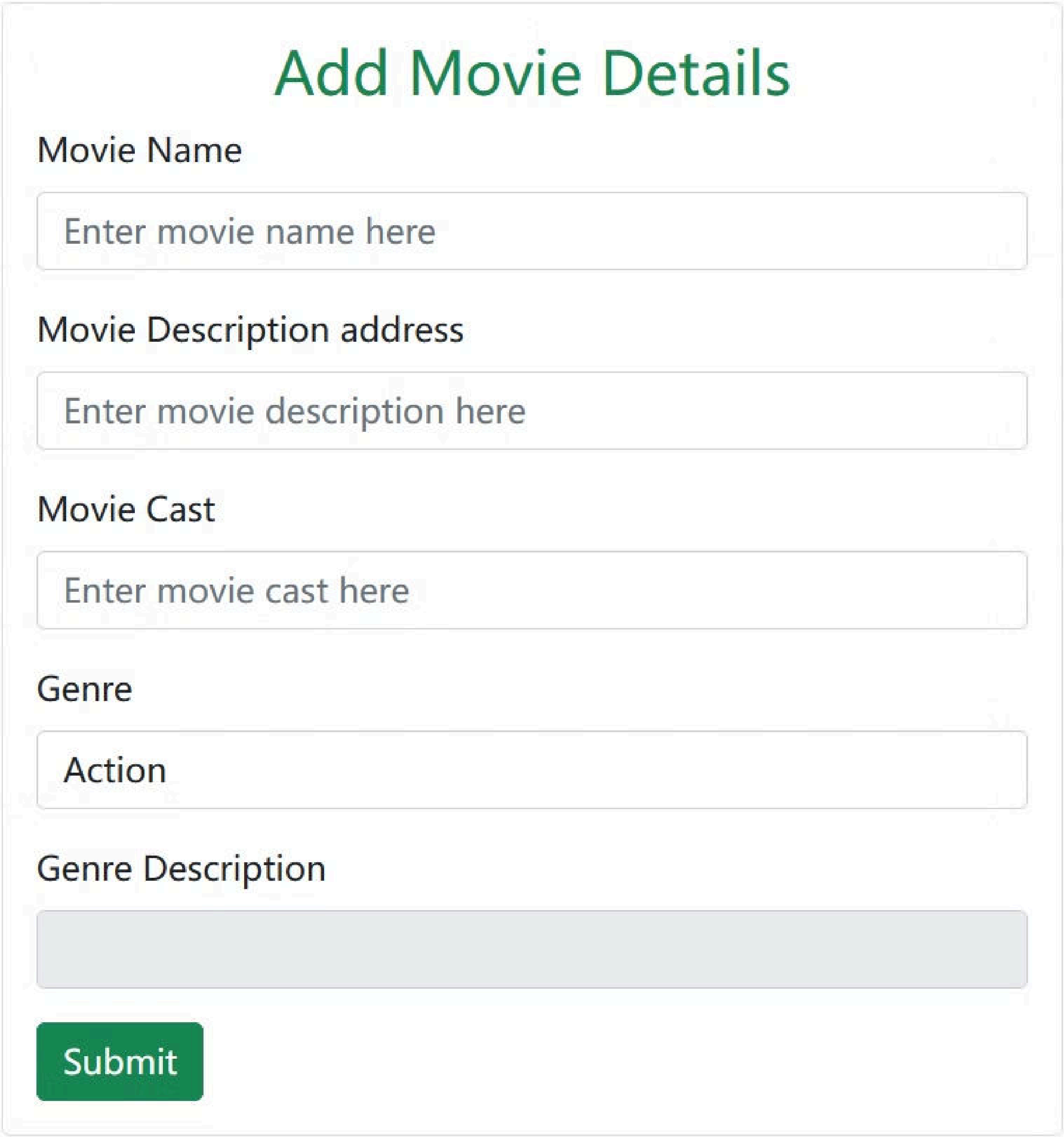
Movie Name
The Matrix

Movie Description
A computer hacker learns from mysterious rebels about th

Cast
Keanu Reeves, Laurence Fishburne

Genre

Genre Description
A genre that emphasizes physical feats, including fights, c



Add Movie Details

Movie Name
Enter movie name here

Movie Description address
Enter movie description here

Movie Cast
Enter movie cast here

Genre
Action

Genre Description

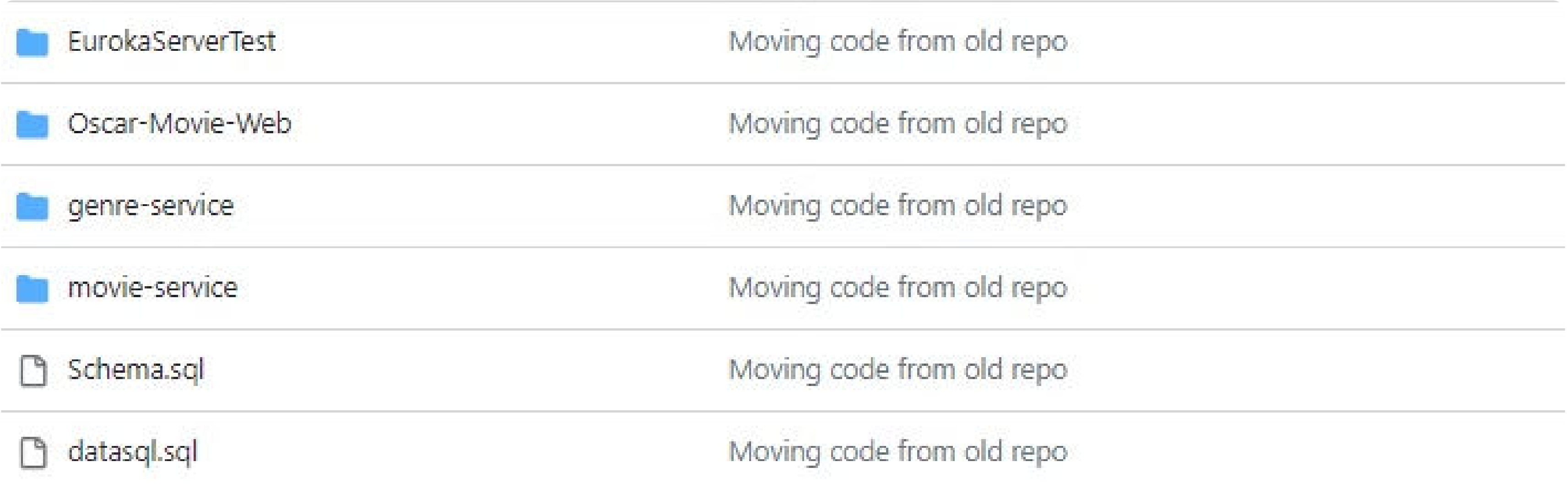
Deployment Instructions

Environment

- Java 17

Installation

git https://github.com/PaulzSh/Oscar-Movie-Data.git



📁 EurokaServerTest	Moving code from old repo
📁 Oscar-Movie-Web	Moving code from old repo
📁 genre-service	Moving code from old repo
📁 movie-service	Moving code from old repo
📄 Schema.sql	Moving code from old repo
📄 datasql.sql	Moving code from old repo

Database

You may run the **schema.sql** follow by **datasql.sql** for the Database setup.

To build the application

Perform **mvnw clean install** command in each project folder.

Access the application

Movie API: http://localhost:8081/swagger-ui/index.html#/

Movie Genre API: http://localhost:8080/swagger-ui/index.html#/

Movie Web Application: http://localhost:8083/

Eureka Server: http://localhost:8761/