Software documentation : IMDB Mini

Spring REST + React

Release 1.0

**Content**

1.About

2.MySQL Database

3. Spring Boot RESTful Web service

Project structure

DAO s

GET/POST Requests

Movies Controller

User Controller

Comments Controller

4. User interface – React

Movie List

Login

Movie page

Register

Search

1. About

IMBD mini is a web application,its main purpose is to make it easy for users to search for movies. Users with admin privilegies can delete,edit or insert movies(DELETE AND EDIT IS NOT IMPLEMENTED).

IMDB mini is a 3-layered application and contains:

MySQL database Server

Spring Boot RESTful Web service

User interface - React

2.MySQL Database

MySQL Database is contained out of 5 tables:

User

user\_id PK

email

username

password

token

group\_id FK

Group

group\_id PK

name

Movie

movie\_id PK

name

date\_of\_release

rating

image\_url

Comment

comment\_id PK

text

user\_id

date\_posted

Movie\_Comment

movie\_comment\_id PK

movie\_id

comment\_id

user\_id

3.Spring Boot RESTful Web service

Project structure

A screenshot of a cell phone

Description automatically generated

DAO s

Comment DAO

A screenshot of a cell phone

Description automatically generated

Movie\_Comment DAO

A screenshot of a cell phone

Description automatically generated

Movie DAO

A screenshot of a cell phone

Description automatically generated

User DAO

A screenshot of a cell phone

Description automatically generated

Movies Controller

Requests:

GET <http://localhost:8082/movies/all> Retrieves all movie

A picture containing screen, table, holding, phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

GET [http://localhost:8082/movies/id/{id}](http://localhost:8082/movies/id/%7bid%7d) Retrieves movie by its id

A close up of a black background

Description automatically generated

A screenshot of a cell phone

Description automatically generated

POST <http://localhost:8082/movies/name> Retrieves movie by its name

A screenshot of a cell phone

Description automatically generated

A black and silver text

Description automatically generated

POST <http://localhost:8082/movies/add> Add movie

A screenshot of a cell phone

Description automatically generated

A picture containing food

Description automatically generated

User Controller

Requests:

GET <http://localhost:8082/users/all> retrieves all users

POST <http://localhost:8082/users/username> retrieves user by his username

POST <http://localhost:8082/users/login> for authentication purposes

POST <http://localhost:8082/users/add> adds the new user

Comments Controller

Requests:

GET [http://localhost:8082/comments/id/{id}](http://localhost:8082/comments/id/%7bid%7d) get all comments by movie id

POST http://localhost:8082/comments/add adds new comment

4.User interface - React

When user joins the web app, he will se a home page that looks like this.

A screenshot of a cell phone

Description automatically generated

Non authenticated user can see Choose between Movie List, Login, Movie page,Register or search for a specific movie

Movie List get all movies from a database(DELETE OR EDIT is not implemented yet, it’s just shown for presentation)

A screenshot of a computer

Description automatically generated

Login form.

A close up of a logo

Description automatically generated

Register Form

A screenshot of a computer screen

Description automatically generated

When user is authenticated his UI looks like this.

A screenshot of a computer screen

Description automatically generated

Current user is admin, so he can add new movies.

Movie page, user can only leave comments if he is logged in.

A screenshot of a computer screen

Description automatically generated