# Movie Catalog Demo Project

Application is intended to expose the rest api for the below

* Movies
* Directors
* Ratings

Framework Used : SpringBoot with the below modules

Web — Full-stack web development with Tomcat and Spring MVC

DevTools — Spring Boot Development Tools

JPA — Java Persistence API including spring-data-JPA, spring-orm, and Hibernate

MySQL — MySQL JDBC driver

Database : Mysql (Please refer the dbscript in the resource section)

**Rest APIs are as below**

**For Director**

http://<hostname:portno>/v1/directors : GET to list directors

http://<hostname:portno>/v1/directors : Post to create the new director

http://<hostname:portno>/v1/directors/{id} : GET to get the director by ID

given in the path variable

http://<hostname:portno>/v1/directors/{id} : DELETE to delete the director

http://<hostname:portno>/v1/directors/{id} : PUT to update the director details provided in the id and body.

**For Movies**

http://<hostname:portno>/v1/movies : GET to list movies

http://<hostname:portno>/v1/movies : Post to create the new movie

http://<hostname:portno>/v1/movies /{id} : GET to get the movie by ID

given in the path variable

http://<hostname:portno>/v1/movies /{id} : DELETE to delete the movie

http://<hostname:portno>/v1/movies /{id} : PUT to update the movie details provided in the id and body.

http://<hostname:portno>/v1/movies : GET with request parameter directorfirstname and directorlastname to search movies based on the director name

(http://localhost:8080/v1/movies?directorfirstname=shruthi&directorlastname=lynch)

http://<hostname:portno>/v1/movies : GET with request parameter rating to search movies which are greater than or equal to the provided rating .

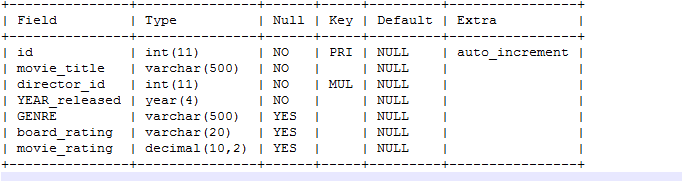
(<http://localhost:8080/api/v1/movies?rating=8.5>)

For the ratings separate rest API are not coded because the rating cannot exist as a separate entity it will be always be associated with the movie

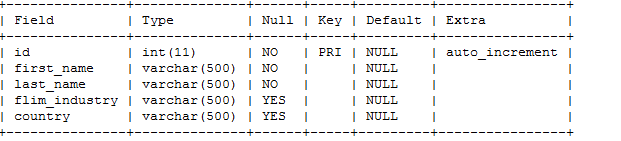
All the rest API returns HTTPSTATUS OK and JSON Body if the calls were successful if not found returns 404 Resource not found status code.

Below 2 Entity are created and mapped into the model

**Movie**

****

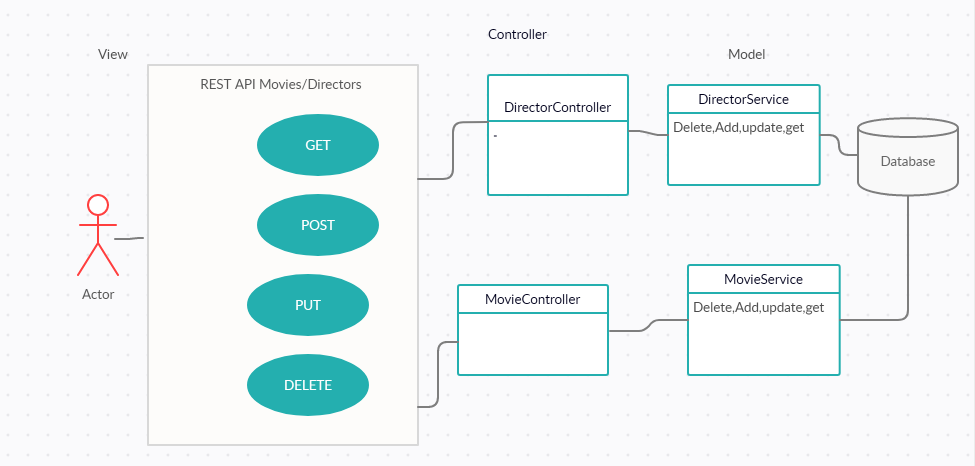
**Director**

****

Junit

Test cases to cover the service layer and the controller layer are written using the junit, mockito and mockmvc to test the restAPI.

**UML Diagram**

****