

EE4216 Lab 4 – RESTful API Design

You are required to submit a lab report to explain the results of each checkpoint with supplementary information such as screenshots and source code as appropriate. A public URL of your website must be provided for each checkpoint for the grader to verify the results.

Objectives:

- Learn to create RESTful API with Spring Boot
- Learn to query data with JdbcTemplate

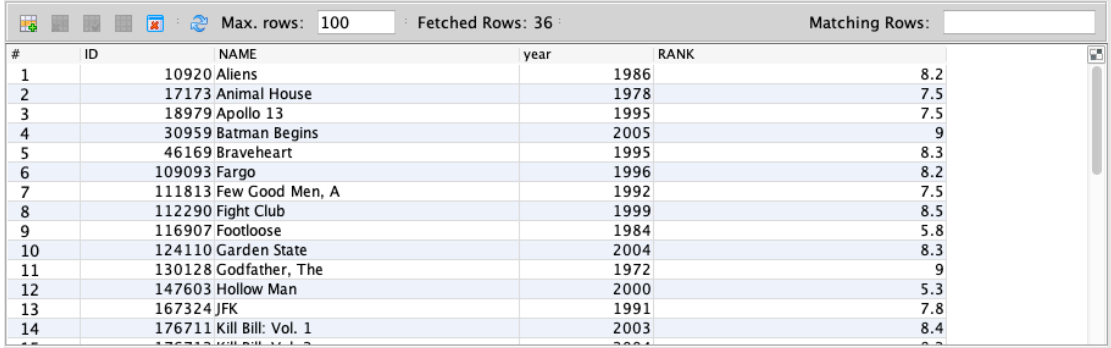
RESTful API is an architecture neutral mean for systems to communicate with each other across the Internet. In previous lab, you have built a dynamic data table using Vue.js and the stub endpoint provided by JSONPlaceholder. This time, you will be guided to build your own RESTful API endpoints to be integrated with the Vue data table.

Before start, you need the following things:

- Your Vue data table client from Lab 3
- The *imdb_small.derby.sql* database script file (download from Canvas)

Task 1 – Importing IMDB Database

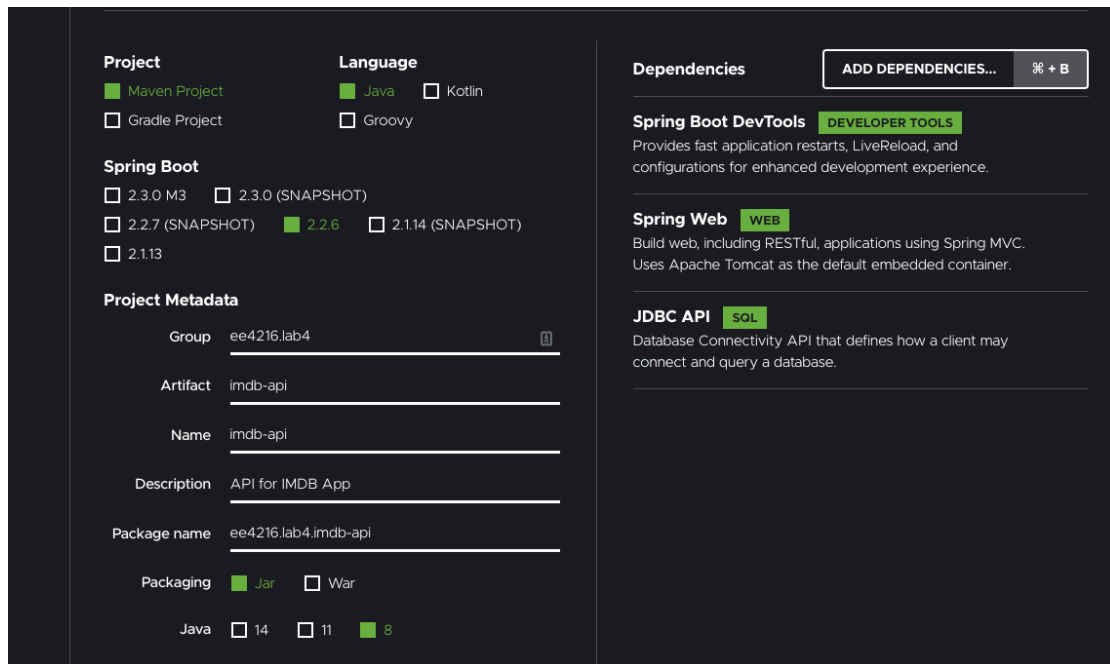
The first step is to set up your database. Start your Java DB server and create a database namely *imdb*, then execute the *imdb_small* script file. You should find 8 tables created under the *imdb* database. The *movies* table looks like this:



#	ID	NAME	year	RANK
1	10920	Aliens	1986	8.2
2	17173	Animal House	1978	7.5
3	18979	Apollo 13	1995	7.5
4	30959	Batman Begins	2005	9
5	46169	Braveheart	1995	8.3
6	109093	Fargo	1996	8.2
7	111813	Few Good Men, A	1992	7.5
8	112290	Fight Club	1999	8.5
9	116907	Footloose	1984	5.8
10	124110	Garden State	2004	8.3
11	130128	Godfather, The	1972	9
12	147603	Hollow Man	2000	5.3
13	167324	JFK	1991	7.8
14	176711	Kill Bill: Vol. 1	2003	8.4

Task 2 – Creating RESTful API Endpoints

Next, generate a Maven project using the Spring Initializr - <https://start.spring.io/>. Add your project dependencies such as the Spring Boot Web stack and the JDBC API as shown in the configuration below. There is no Derby driver can be selected from the Spring Initializr. So, you need to manually add the *derbyclient-10.14* (other versions could be used) dependency to your pom.xml.



The screenshot shows the Spring Initializr configuration page. On the left, under 'Project', 'Maven Project' is selected. Under 'Language', 'Java' is selected. Under 'Spring Boot', '2.2.6' is selected. Under 'Project Metadata', the 'Group' is 'ee4216.lab4', 'Artifact' is 'imdb-api', 'Name' is 'imdb-api', 'Description' is 'API for IMDB App', and 'Package name' is 'ee4216.lab4.imdb-api'. Under 'Packaging', 'Jar' is selected. Under 'Java', '8' is selected. On the right, under 'Dependencies', 'Spring Boot DevTools' (Developer Tools) and 'Spring Web' (Web) are selected. 'JDBC API' (SQL) is also listed but not selected.

Then, create an RESTful API endpoint to return a list of movies from the imdb database in JSON format.

- Create a domain object class *Movie* to represent a movie entity.
- Create a REST controller and map the URL `/api/movies` to your handler method.

A segment of the JSON output is shown as follow:

```
1 [
2   {
3     "id": 10920,
4     "name": "Aliens",
5     "year": 1986,
6     "rank": 8.2
7   },
8   {
9     "id": 17173,
10    "name": "Animal House",
11    "year": 1978,
12    "rank": 7.5
13  },
14 ]
```

Checkpoint 1:

- Explain which HTTP method (GET/POST/UPDATE/DELETE) should be used for the endpoint.
- Explain the code of your *Movie* class and controller class.
- Provide a public URL to demonstrate the JSON response from your API endpoint.

Task 3 – Integrating with Vue Client

After verifying the JSON content as shown above, you can integrate your Vue data table into the Spring Boot project.

- Put your Vue client's files (html/js/css) in the /resources/static directory.
- Update the relevant URLs pointing to the API endpoint.
- Update your data table schema as appropriate.

The final result looks like this:

ID	Movie Name ^	Year	Ranking
10920	Aliens	1986	8.2
17173	Animal House	1978	7.5
18979	Apollo 13	1995	7.5
30959	Batman Begins	2005	9
46169	Braveheart	1995	8.3
109093	Fargo	1996	8.2
111813	Few Good Men, A	1992	7.5
112290	Fight Club	1999	8.5
116907	Footloose	1984	5.8

Sorted by movie name

ID	Movie Name	Year ^	Ranking
130128	Godfather, The	1972	9
313459	Star Wars	1977	8.8
17173	Animal House	1978	7.5
116907	Footloose	1984	5.8
10920	Aliens	1986	8.2
257264	Planes, Trains & Automobiles	1987	7.2
192017	Little Mermaid, The	1989	7.3
344203	UHF	1989	6.6
167324	JFK	1991	7.8

Sorted by year

Checkpoint 2:

- Explain how you deploy and run the application on a public server.
- Provide a public URL to demonstrate the result. Explain the Vue code that you have modified.

- END -