# PRACTICAL ON CLOUD NATIVE APPLICATION

---

**Name:** Atharva Paliwal

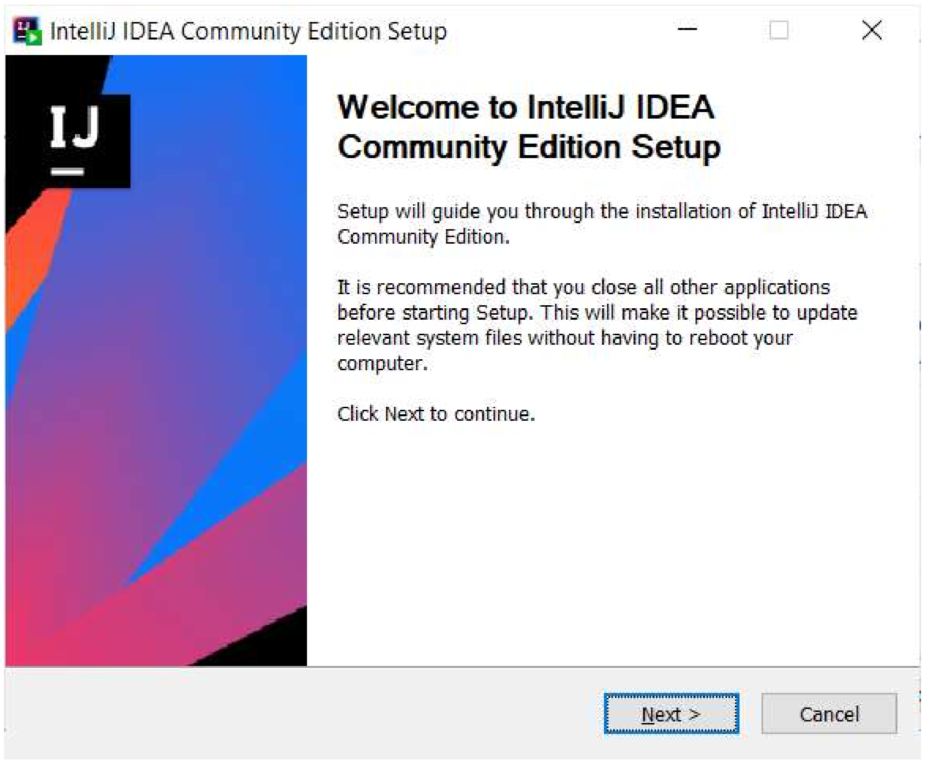
**Roll No.:** B40

---

**Problem Statement:** To develop an application using cloud native approach with microservice technique.

**OUTPUT**

1. Install an IDE - IntelliJ IDEA

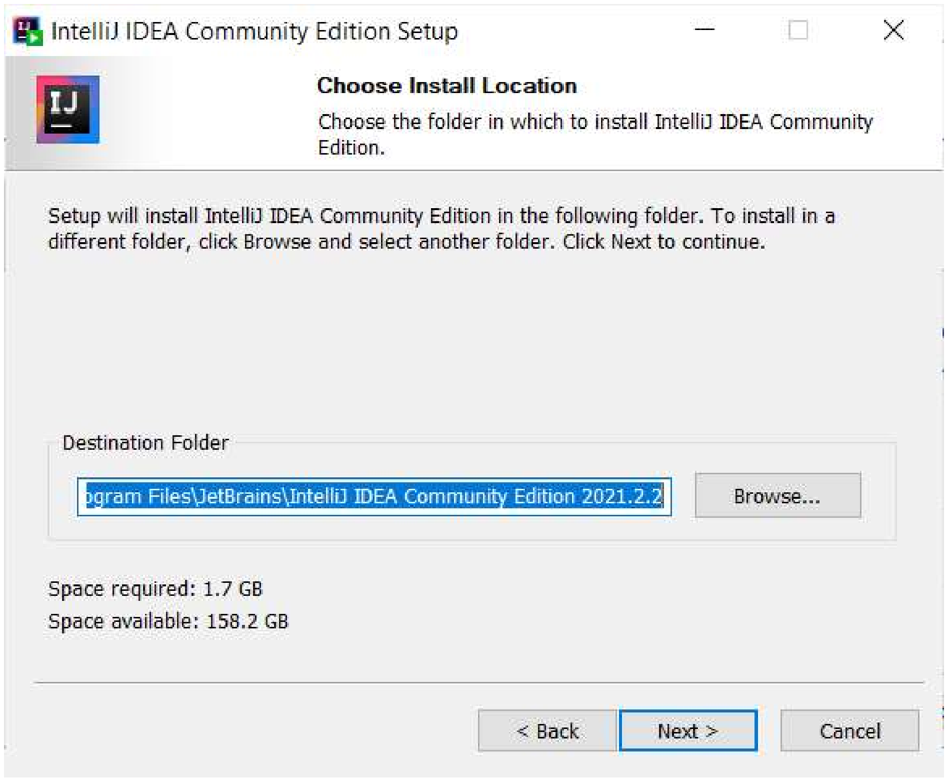

IntelliJ IDEA Community Edition Setup

Welcome to IntelliJ IDEA Community Edition Setup

Setup will guide you through the installation of IntelliJ IDEA Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.


IntelliJ IDEA Community Edition Setup

**Choose Install Location**

Choose the folder in which to install IntelliJ IDEA Community Edition.

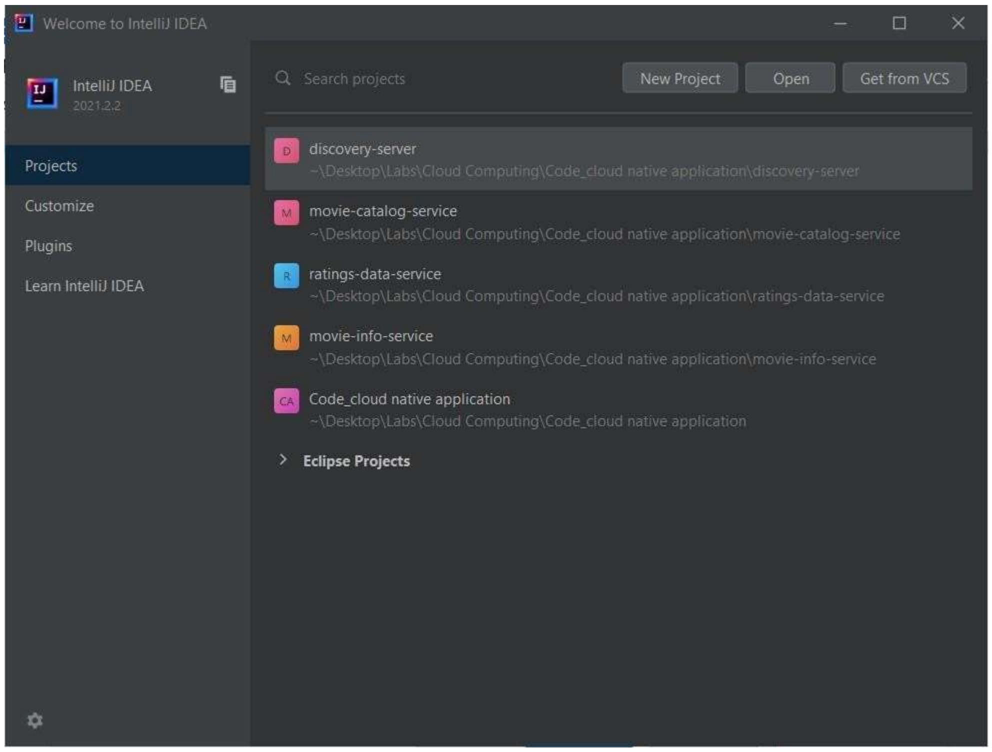
Setup will install IntelliJ IDEA Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

ogram Files\JetBrains\IntelliJ IDEA Community Edition 2021.2.2

Space required: 1.7 GB

Space available: 158.2 GB


Welcome to IntelliJ IDEA

IntelliJ IDEA 2021.2.2

discovery-server
~\Desktop\Labs\Cloud Computing\Code_cloud native application\discovery-server

movie-catalog-service
~\Desktop\Labs\Cloud Computing\Code_cloud native application\movie-catalog-service

ratings-data-service
~\Desktop\Labs\Cloud Computing\Code_cloud native application\ratings-data-service

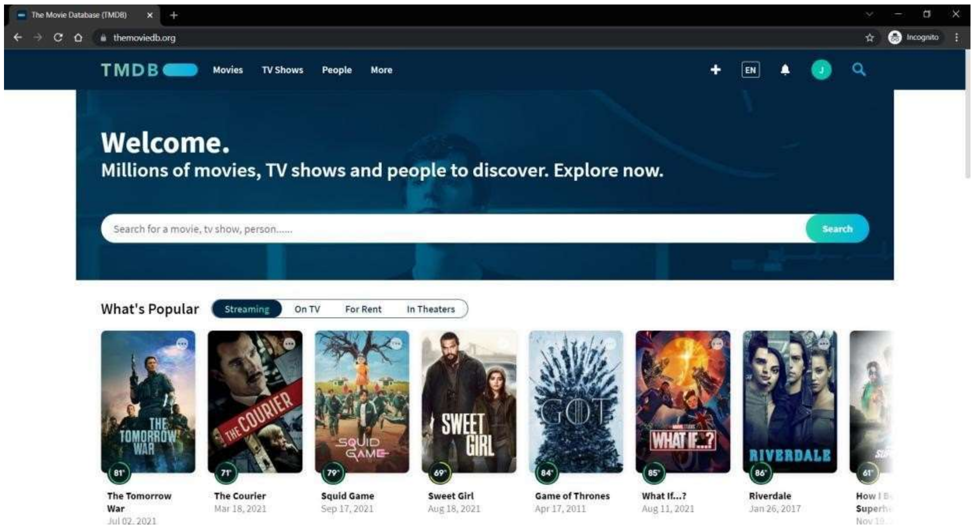
movie-info-service
~\Desktop\Labs\Cloud Computing\Code_cloud native application\movie-info-service

Code_cloud native application
~\Desktop\Labs\Cloud Computing\Code_cloud native application

Eclipse Projects

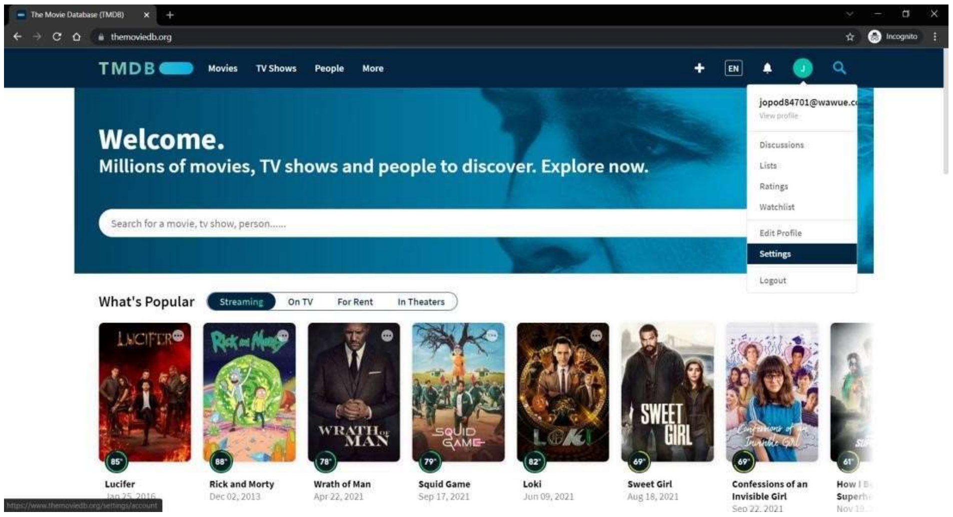
2. Add API key from MovieDB website

   a) Open given link: https://www.themoviedb.org/

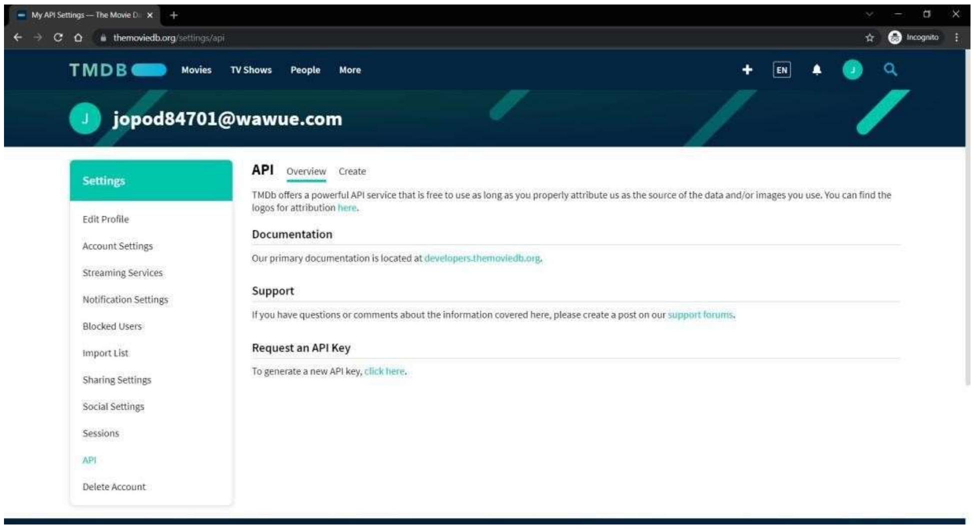

TMDB
Movies
TV Shows
People
More
Welcome.
Millions of movies, TV shows and people to discover. Explore now.
Search for a movie, tv show, person......
Search
What's Popular
Streaming
On TV
For Rent
In Theaters

   b) Sign up if account is not created and then do login

   c) Click on Setting option as shown in figure below:



d) Go to API section



**API**

TMDb offers a powerful API service that is free to use as long as you properly attribute us as the source of the data and/or images you use. You can find the logos for attribution here.

**Documentation**

Our primary documentation is located at developers.themoviedb.org.

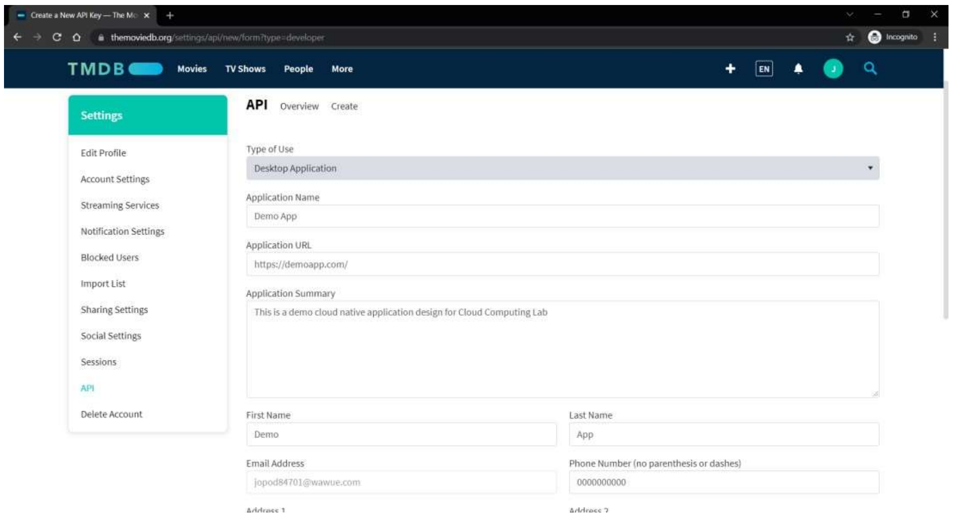
**Support**

If you have questions or comments about the information covered here, please create a post on our support forums.

**Request an API Key**

To generate a new API key, click here.

e) This will ask for details regarding the application that you are developing. Add the details then it will generate API key for you.



**API**

Type of Use: Desktop Application

Application Name: Demo App

Application URL: https://demoapp.com/

Application Summary: This is a demo cloud native application design for Cloud Computing Lab

First Name: Demo

Last Name: App

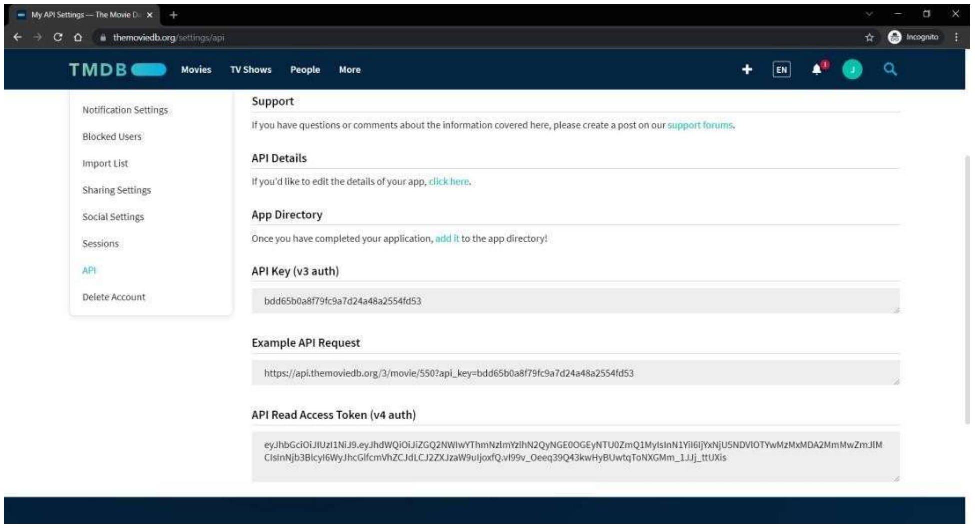
Email Address: jopod84701@wawue.com

Phone Number (no parenthesis or dashes): 0000000000

Address 1

Address 2

f) Copy the generated API key



**Support**

If you have questions or comments about the information covered here, please create a post on our support forums.

**API Details**

If you'd like to edit the details of your app, click here.

**App Directory**

Once you have completed your application, add it to the app directory!

**API Key (v3 auth)**

```
bdd65b0a8f79fc9a7d24a48a2554fd53
```

**Example API Request**

```
https://api.themoviedb.org/3/movie/550?api_key=bdd65b0a8f79fc9a7d24a48a2554fd53
```

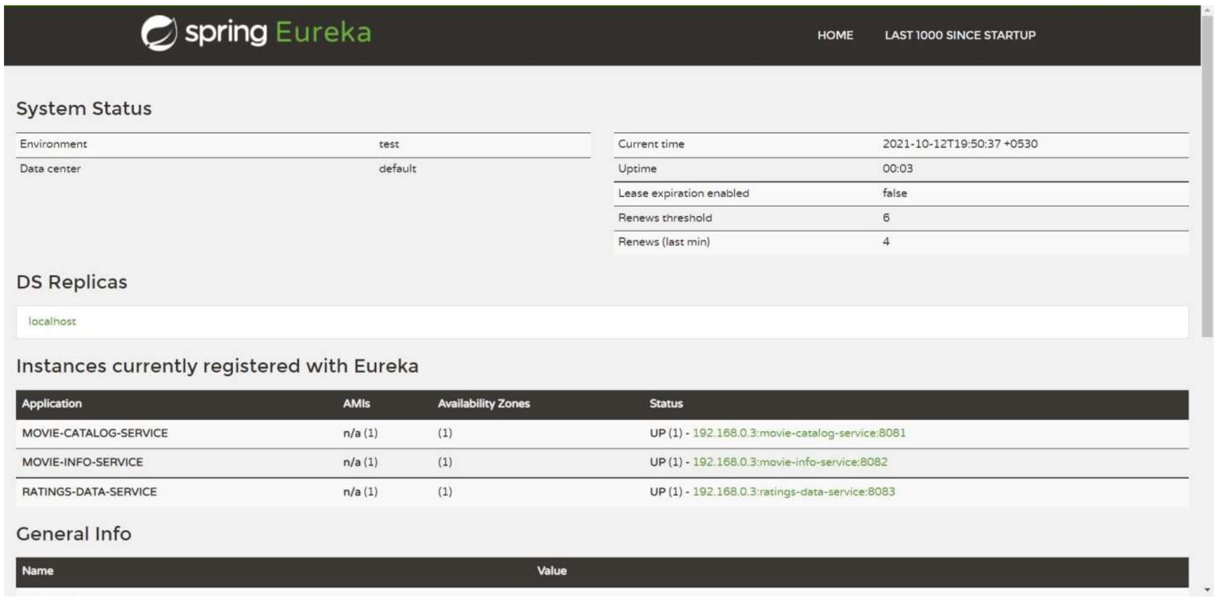
**API Read Access Token (v4 auth)**

```
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJiZGQ2NWIwYThmNzlmYzlhN2QyNGE0OGEyNTU0ZmQ1MyIsInN1YiI6IjYxNjU5NDVlOTYwMzMxMDA2MmMwZmJlM
CIsInNjb3BlcyI6WyJhcGlfcmVhZCJdLCJ2ZXJzaW9uIjoxfQ.vI99v_Oeeq39Q43kwHyBUwtqToNXGMm_1JJj_ttUXis
```

g) Add this API key in application. Properties in movie-info- service component

## Implementation

1. **Eureka server** - This Eureka Server contains all the details regarding the components that are running of a particular application and also shows thenumber of instances that are running for each service.



System Status

| | |
|---|---|
| Environment | test |
| Data center | default |

| | |
|---|---|
| Current time | 2021-10-12T19:50:37 +0530 |
| Uptime | 00:03 |
| Lease expiration enabled | false |
| Renews threshold | 6 |
| Renews (last min) | 4 |

DS Replicas

localhost

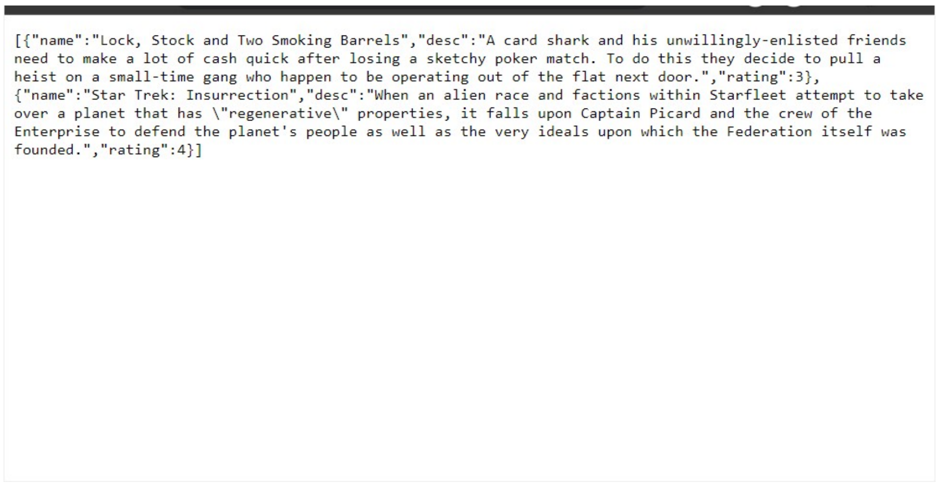
Instances currently registered with Eureka

| Application | AMIs | Availability Zones | Status |
|---|---|---|---|
| MOVIE-CATALOG-SERVICE | n/a (1) | (1) | UP (1) - 192.168.0.3:movie-catalog-service:8081 |
| MOVIE-INFO-SERVICE | n/a (1) | (1) | UP (1) - 192.168.0.3:movie-info-service:8082 |
| RATINGS-DATA-SERVICE | n/a (1) | (1) | UP (1) - 192.168.0.3:ratings-data-service:8083 |

General Info

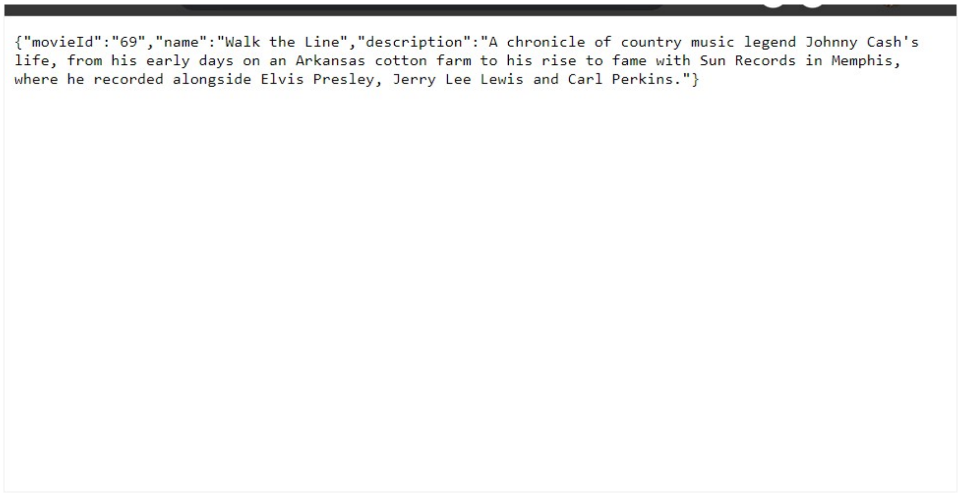
| Name | Value |
|---|---|

2. The **movie-catalog-service** can be run which will call a certain API and it displays the information regarding the movie and its rating entered by that particular user. Output is displayed in localhost. Simply write localhost/port-no/user ID where port-no is the port number allocated to run this particular service and user ID is the Id of the user for whom we want to know the details.



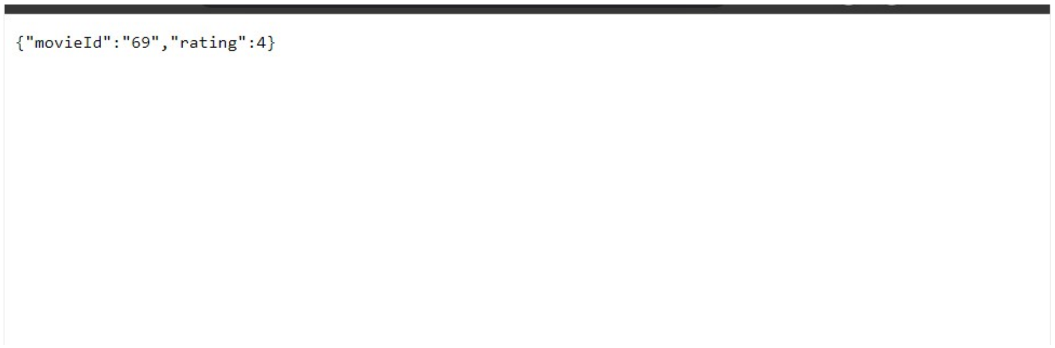
```
[{"name":"Lock, Stock and Two Smoking Barrels","desc":"A card shark and his unwillingly-enlisted friends
need to make a lot of cash quick after losing a sketchy poker match. To do this they decide to pull a
heist on a small-time gang who happen to be operating out of the flat next door.","rating":3},
{"name":"Star Trek: Insurrection","desc":"When an alien race and factions within Starfleet attempt to take
over a planet that has \"regenerative\" properties, it falls upon Captain Picard and the crew of the
Enterprise to defend the planet's people as well as the very ideals upon which the Federation itself was
founded.","rating":4}]
```

3. The **movie-info-service** can be run which will call a certain API and it displays the information regarding the movie. Output is displayed in localhost. It shows the movie id, name of the movie, and it’s description.



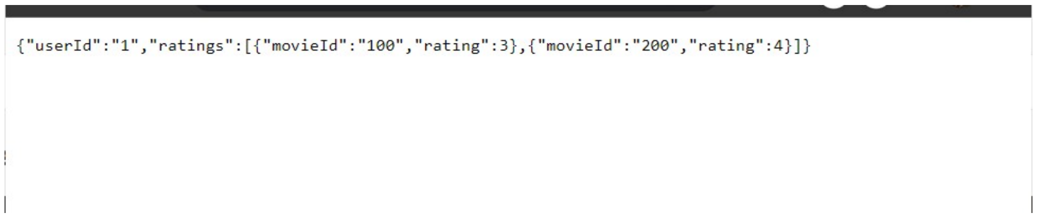
```
{"movieId":"69","name":"Walk the Line","description":"A chronicle of country music legend Johnny Cash's
life, from his early days on an Arkansas cotton farm to his rise to fame with Sun Records in Memphis,
where he recorded alongside Elvis Presley, Jerry Lee Lewis and Carl Perkins."}
```

4. The **ratings-data-service** can be run which will call a certain API and it displays the information regarding the number of ratings a movie has received. Output is displayed in localhost.



```
{"movieId":"69","rating":4}
```

5. The **ratings-data-service** can also be used to display the information regarding a particular user and all the ratings he has given. Output is displayed in localhost.



```
{"userId":"1","ratings":[{"movieId":"100","rating":3},{"movieId":"200","rating":4}]}
```

***** END ****