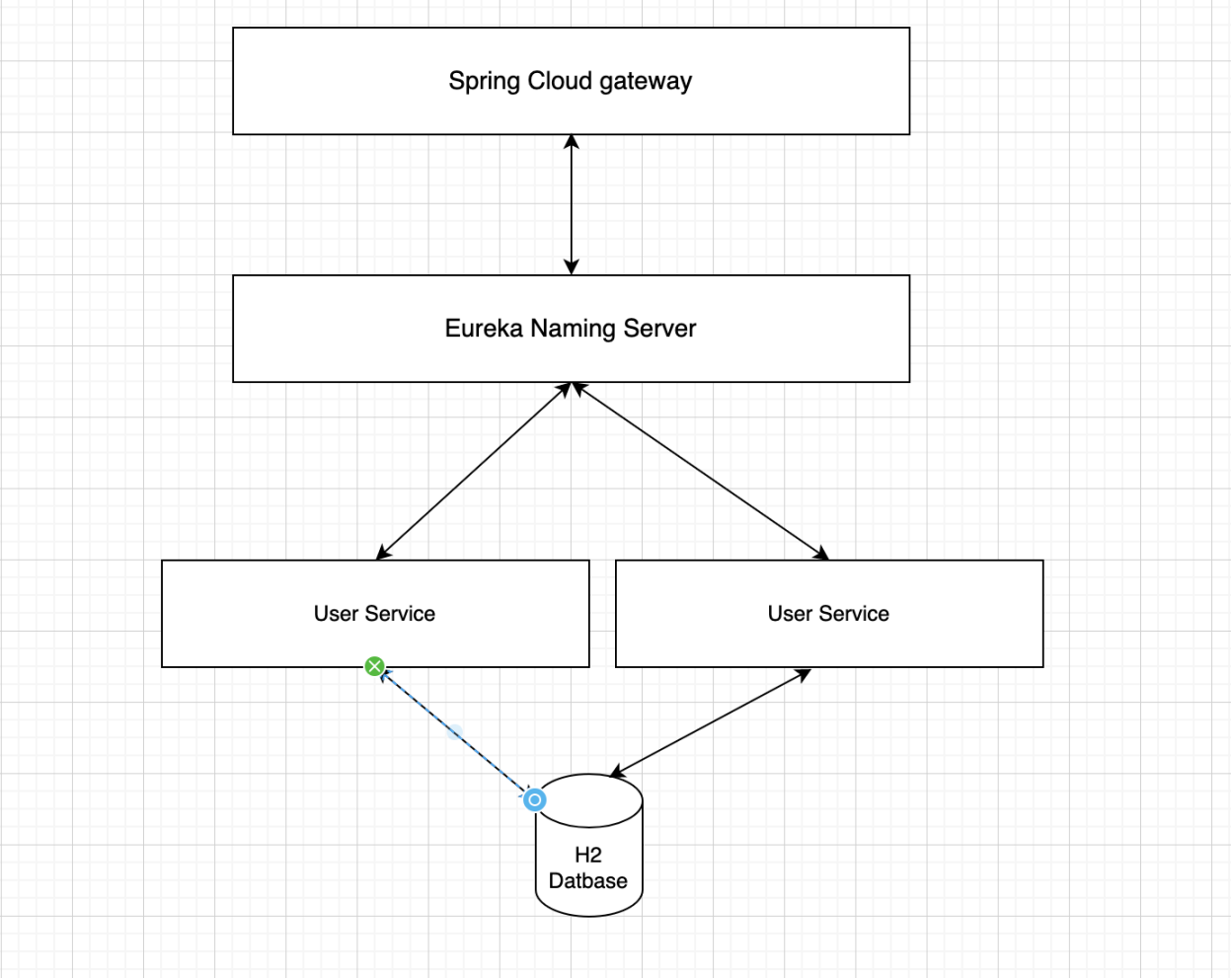
Problem Statement
Feature: Develop a Restful API to Create, Update, and Delete Users
Users shall be stored in a database (For example H2 in-memory database)

I have used Spring Boot and Spring Cloud for developing the user service.

Architecture



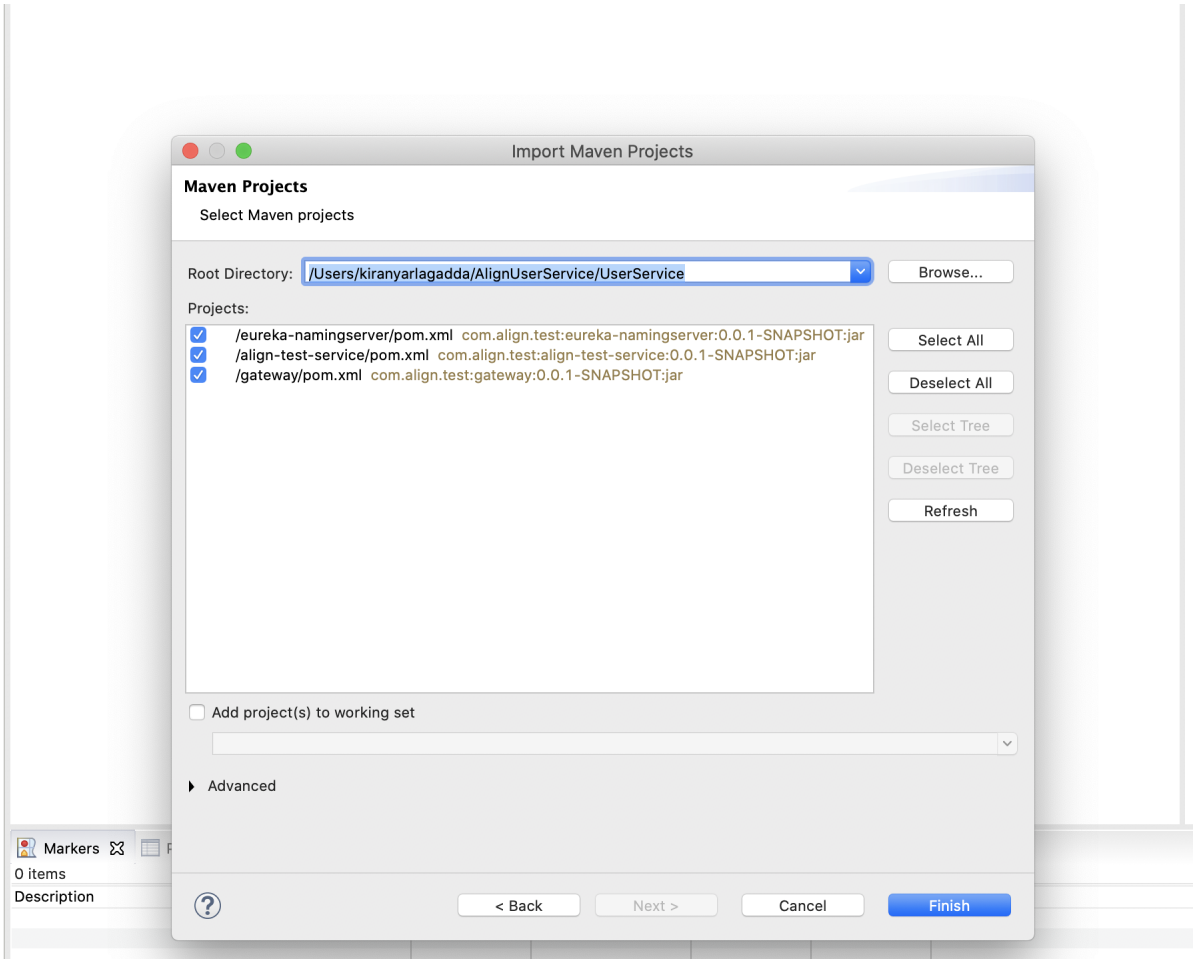
Spring Cloud Gateway is used for load balancing and can also be used for authentication and authorization of services along with monitoring,

Eureka Naming Server/ Discovery Server used to register the user services so that these services can be used by gateway.

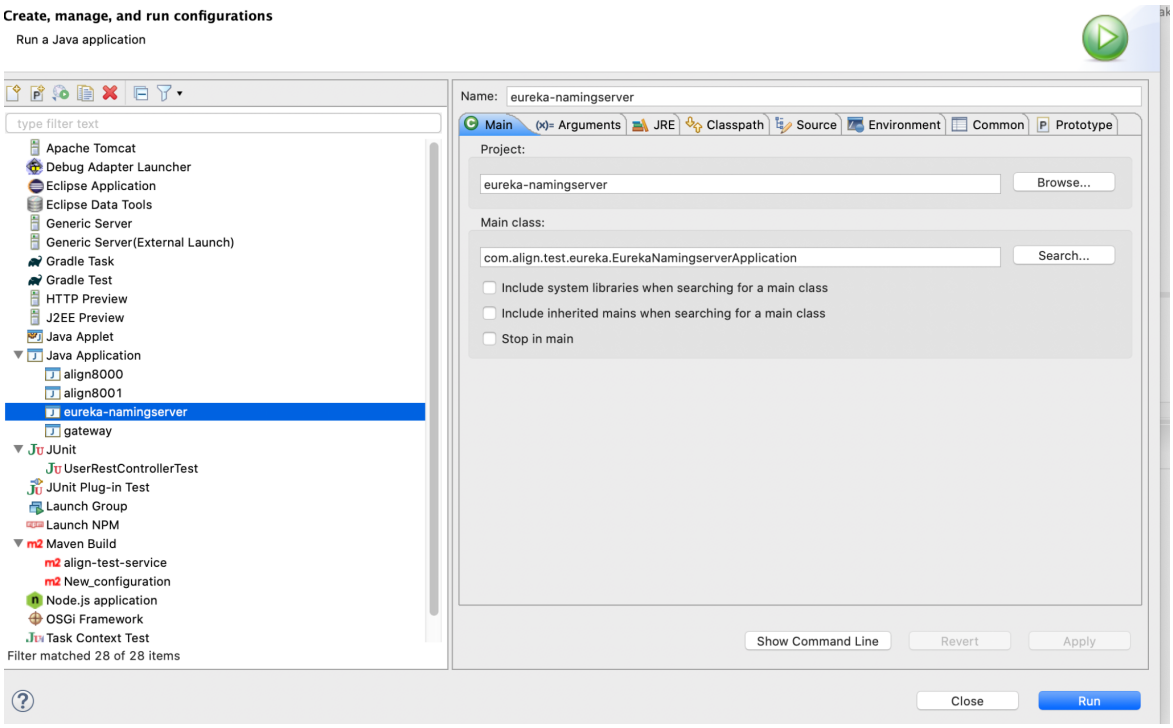
To Test it use the below url to get the code.

git clone https://github.com/kiranycode/UserService.git

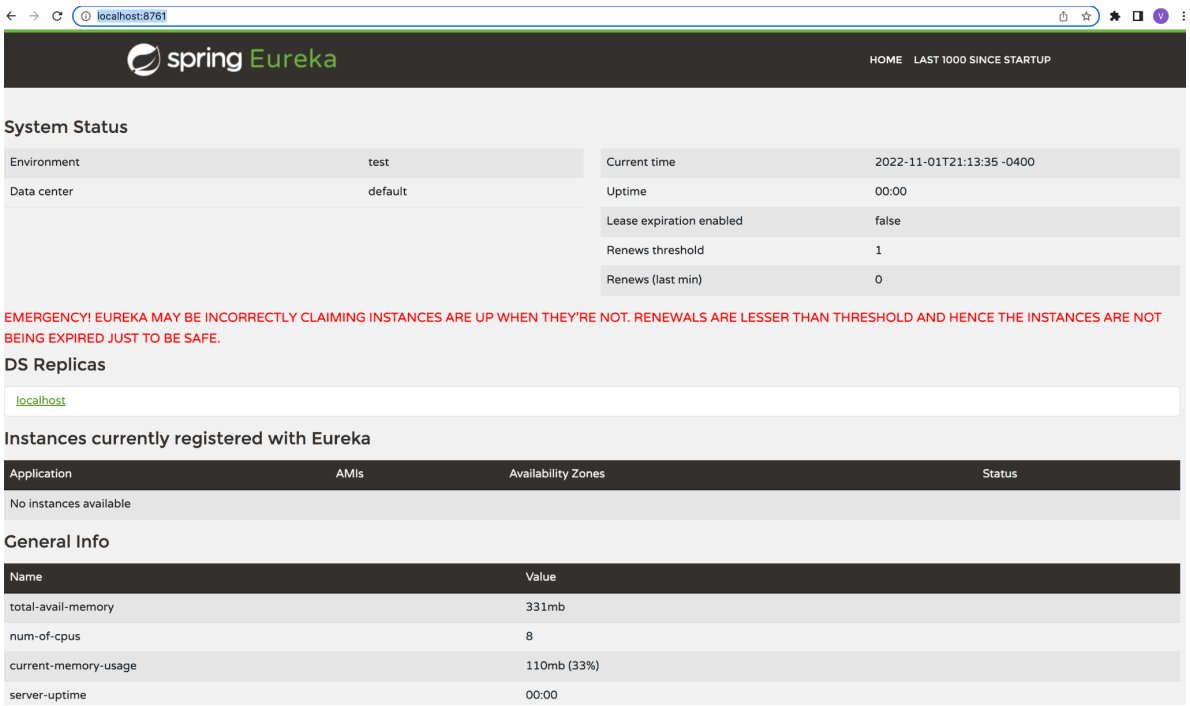
Import the projects into Eclipse using import Existing Maven projects



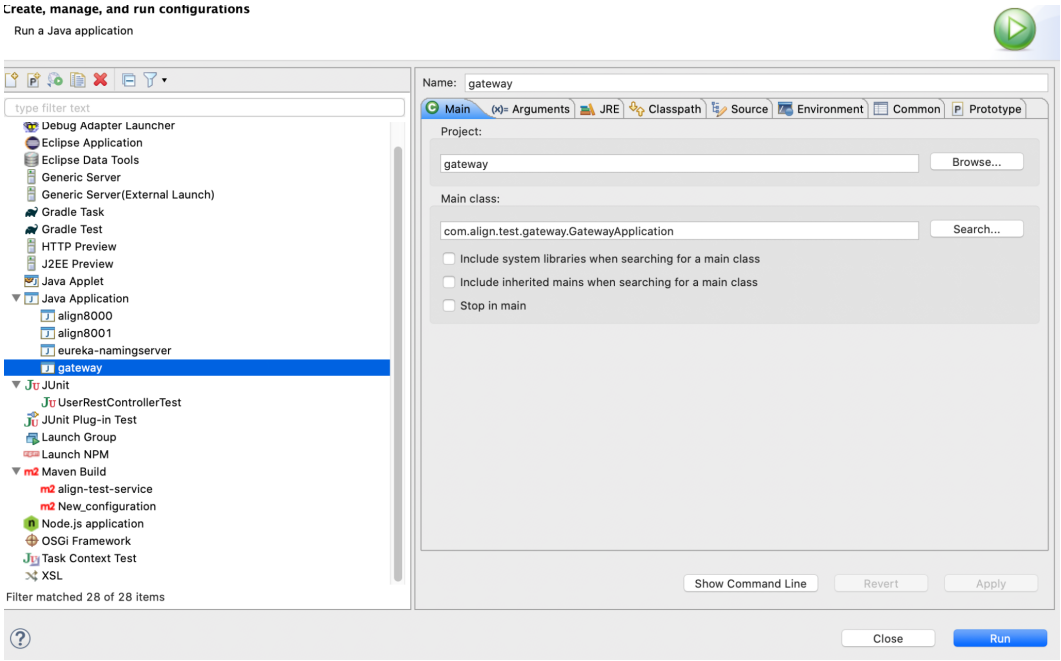
Start the naming server from eclipse as below



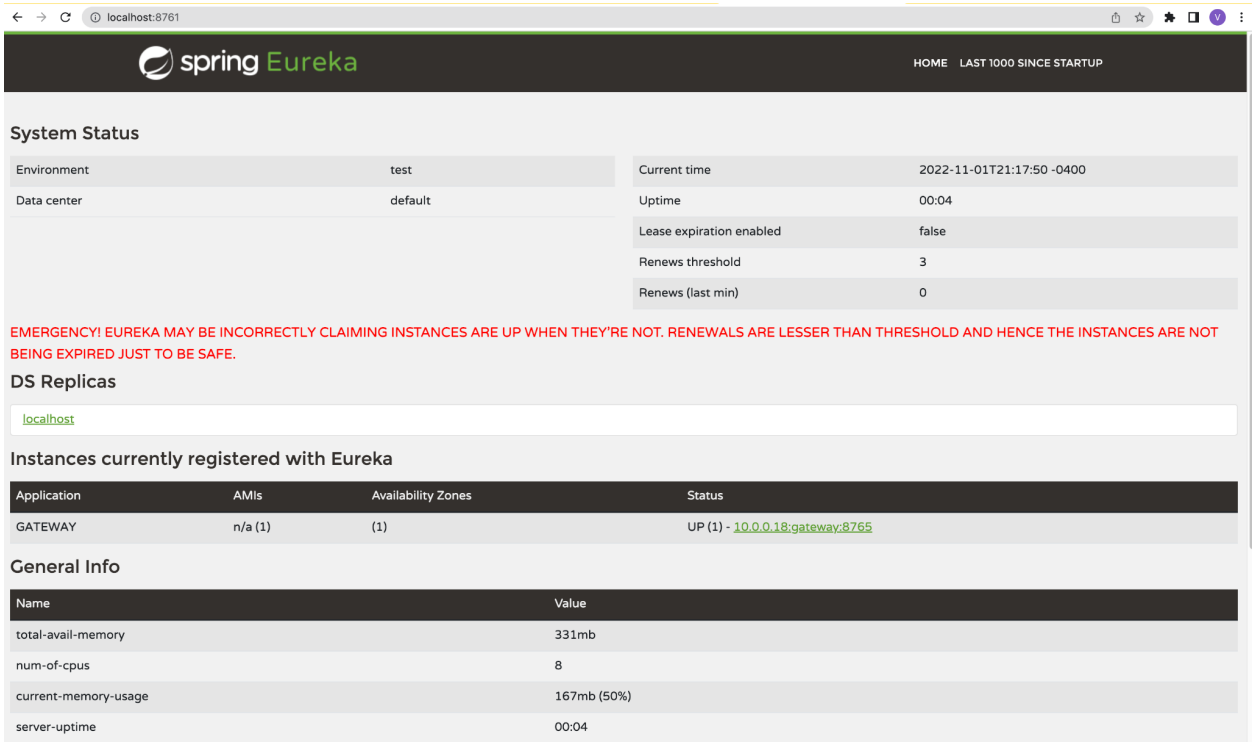
Access naming server with url:http://localhost:876



Start the gateway server from eclipse



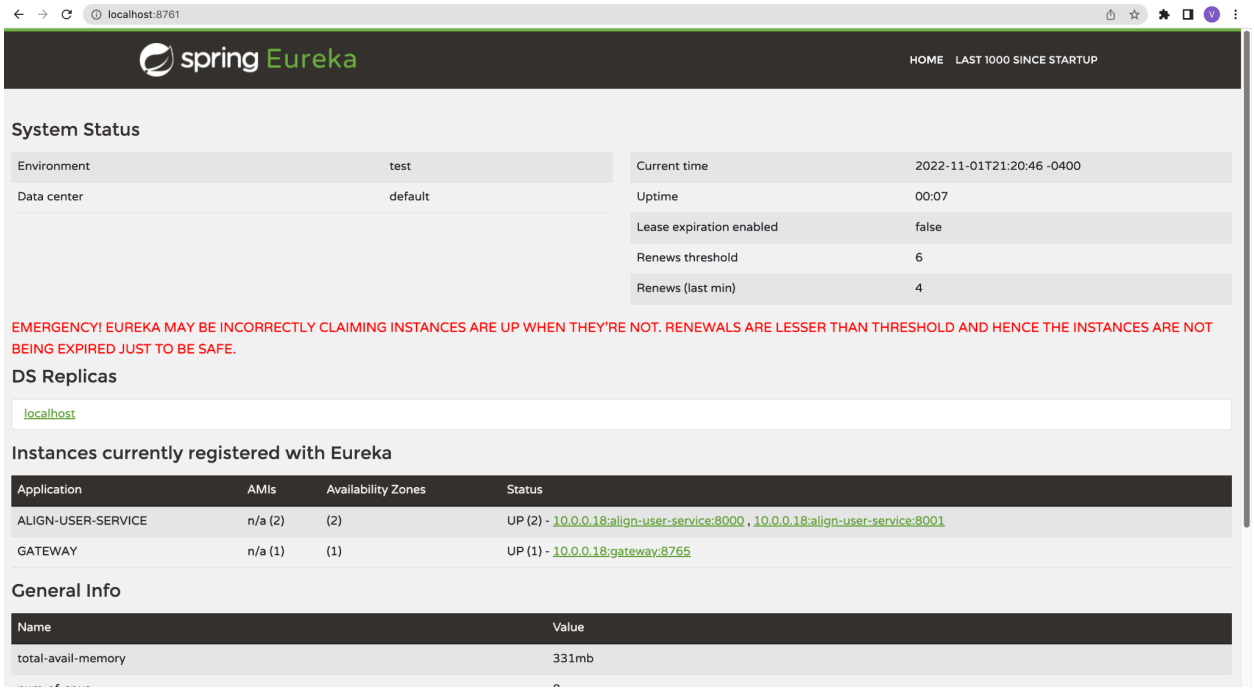
Refresh the naming server url you can see that gateway registers with the naming server as seen below.



Start the userservice on port 8000 and 8001 in eclipse with following vm parameters

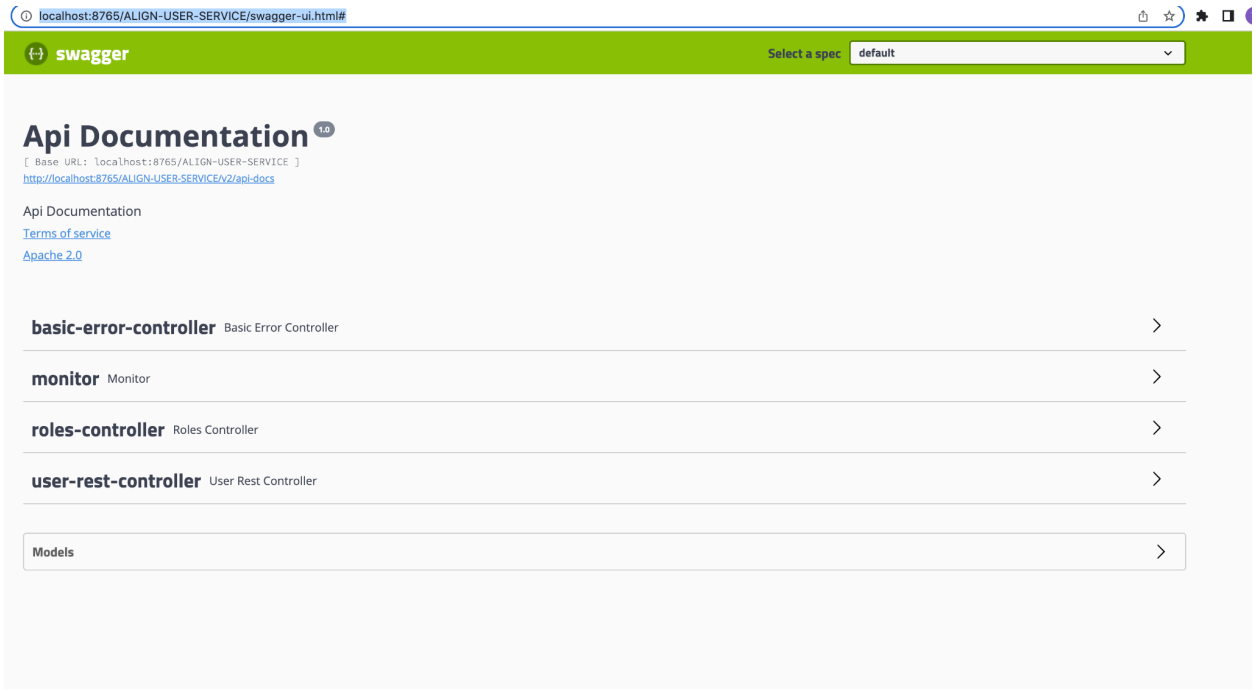
-Dserver.port=8000 -Denv=A and -Dserver.port=8001 -Denv=B

The user services will also register on the naming server.



Access the Swagger UI with below url

http://localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#(via gateway 8765 gateway port)



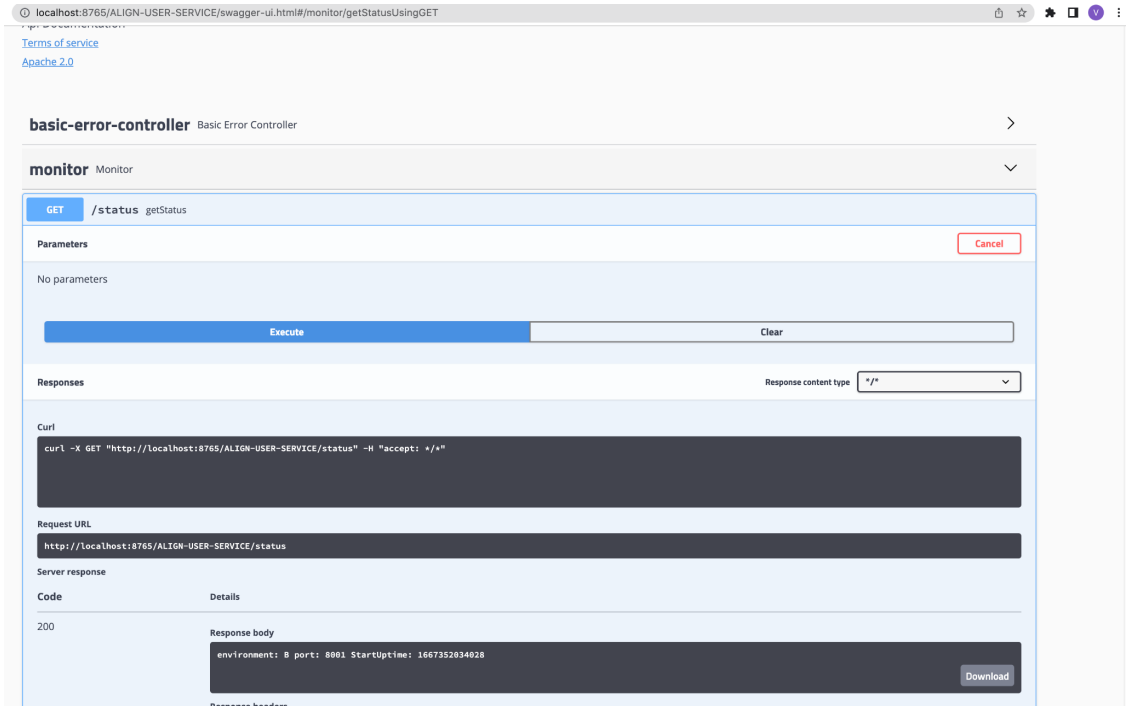
You can check the status and test the services from the swagger ui.

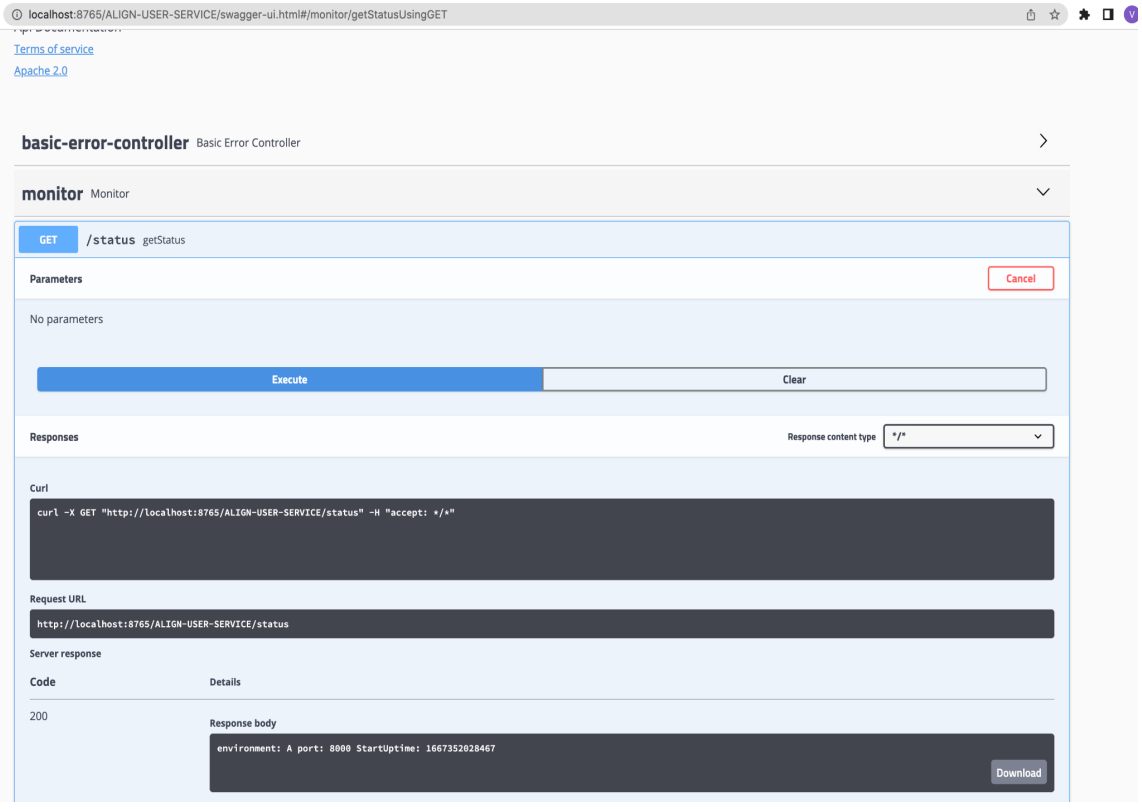
For example to test the status use url http://localhost:8765/ALIGN-USER-SERVICE/status /swagger ui

First time the response is

environment: A port: 8000 StartUptime: 1667352028467 and second time it is environment: B port: 8001 StartUptime: 1667352034028

We can see the load balance happening here.


localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#/monitor/getStatusUsingGET
Terms of service
Apache 2.0
basic-error-controller Basic Error Controller
monitor Monitor
GET /status getStatus
Parameters
Cancel
No parameters
Execute
Clear
Responses
Response content type */*
Curl
curl -X GET "http://localhost:8765/ALIGN-USER-SERVICE/status" -H "accept: */*"
Request URL
http://localhost:8765/ALIGN-USER-SERVICE/status
Server response
Code
Details
200
Response body
environment: B port: 8001 StartUptime: 1667352034028
Download


localhost:8765/ALIGN-USER-SERVICE/swagger-ui.html#/monitor/getStatusUsingGET
Terms of service
Apache 2.0
basic-error-controller Basic Error Controller
monitor Monitor
GET /status getStatus
Parameters
Cancel
No parameters
Execute
Clear
Responses
Response content type */*
Curl
curl -X GET "http://localhost:8765/ALIGN-USER-SERVICE/status" -H "accept: */*"
Request URL
http://localhost:8765/ALIGN-USER-SERVICE/status
Server response
Code
Details
200
Response body
environment: A port: 8000 StartUptime: 1667352028467
Download

You can find some of the request and response for some of the services.

Create Roles

http://localhost:8765/ALIGN-USER-SERVICE/api/roles/create

Request

```
{
  "id": 1,
  "roleName": "admin"
}
```

Create User

http://localhost:8765/ALIGN-USER-SERVICE/api/user/createUser

Request

```
{
  "email": "kk@kk.com",
  "password": "1234",
  "rolesRequest": [
    {
```

```
      "id": 1
    }
  ],
  "userName": "kiran"
}
```


Response


```
{
  "uuid": "90464fb5-8601-47ec-959e-a50bdaa8be40",
  "userName": "kiran",
  "email": "kk@kk.com",
  "password": "1234",
  "rolesResponse": [
    {
      "id": 1,
      "roleName": "admin"
    }
  ]
```

}

Get User By UUID

http://localhost:8765/ALIGN-USER-SERVICE/api/user/getUserById/90464fb5-8601-47ec-959e-a50bdaa8be40

GET User By Email

http://localhost:8765/ALIGN-USER-SERVICE/api/user/getUserByUserEmail/kk%40kk.com

Get User By Name

http://10.0.0.18:8765/ALIGN-USER-SERVICE/api/user/getUserByUserName/kiran

For **Auto-scaling** we can deploy these to any cloud environment and use the monitoring system to monitor and auto scale it.
and we can generate the images by using the maven build **mvn spring-boot:build-image -DskipTests** and use in container world and orchestrate with kubernetes

If you want to generate the image uncomment/add this in pom.xml, **kirany is my dockerhub login**

```
<image>
        <name>kirany/mmv2-${project.artifactId}:${project.version}</name>
  </image>
```