**High Level Architecture Diagram:**

Eureka Discovery Server

User db

User service

Zuul Api Gateway

Book db

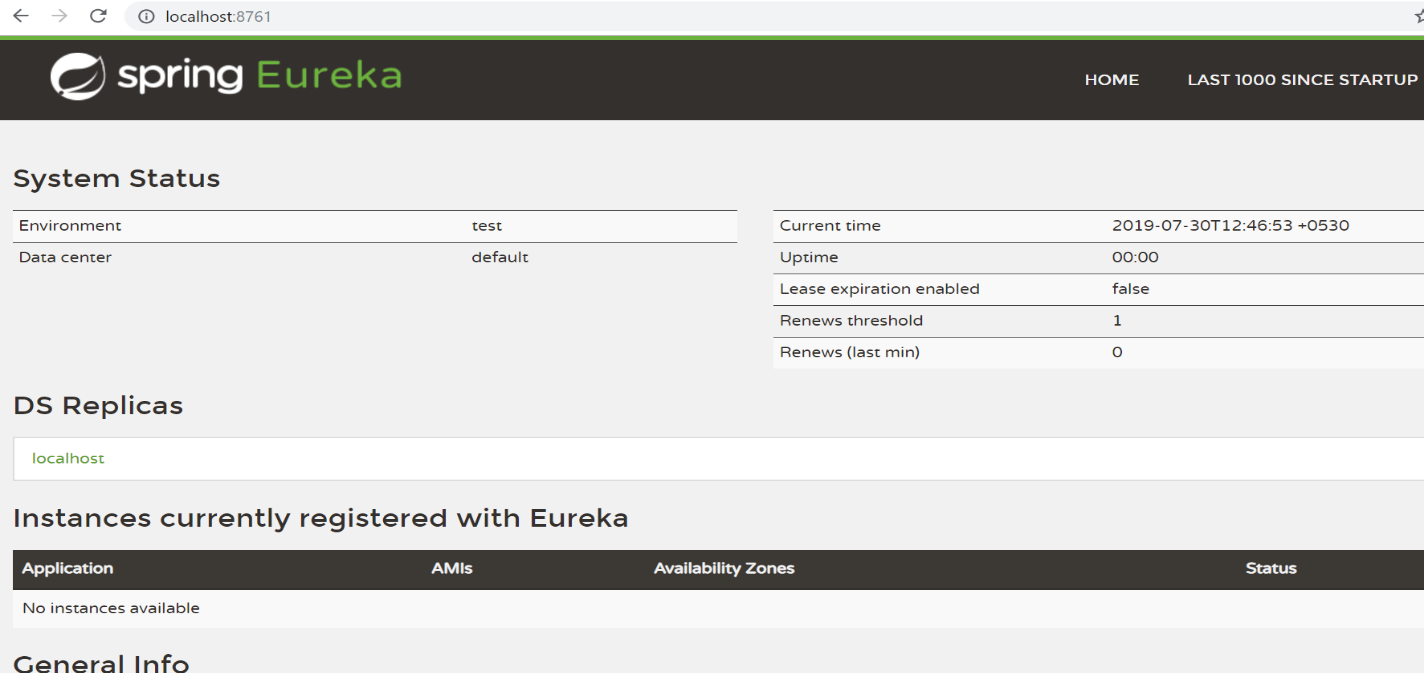
Book service

**Steps to setup an application:**

Step 1: import all the projects in STS.

Step 2: In D drive create two folder one with name userdata and other with name bookdata.

Step 3: start discovery-server project it will be running on port no 8761.

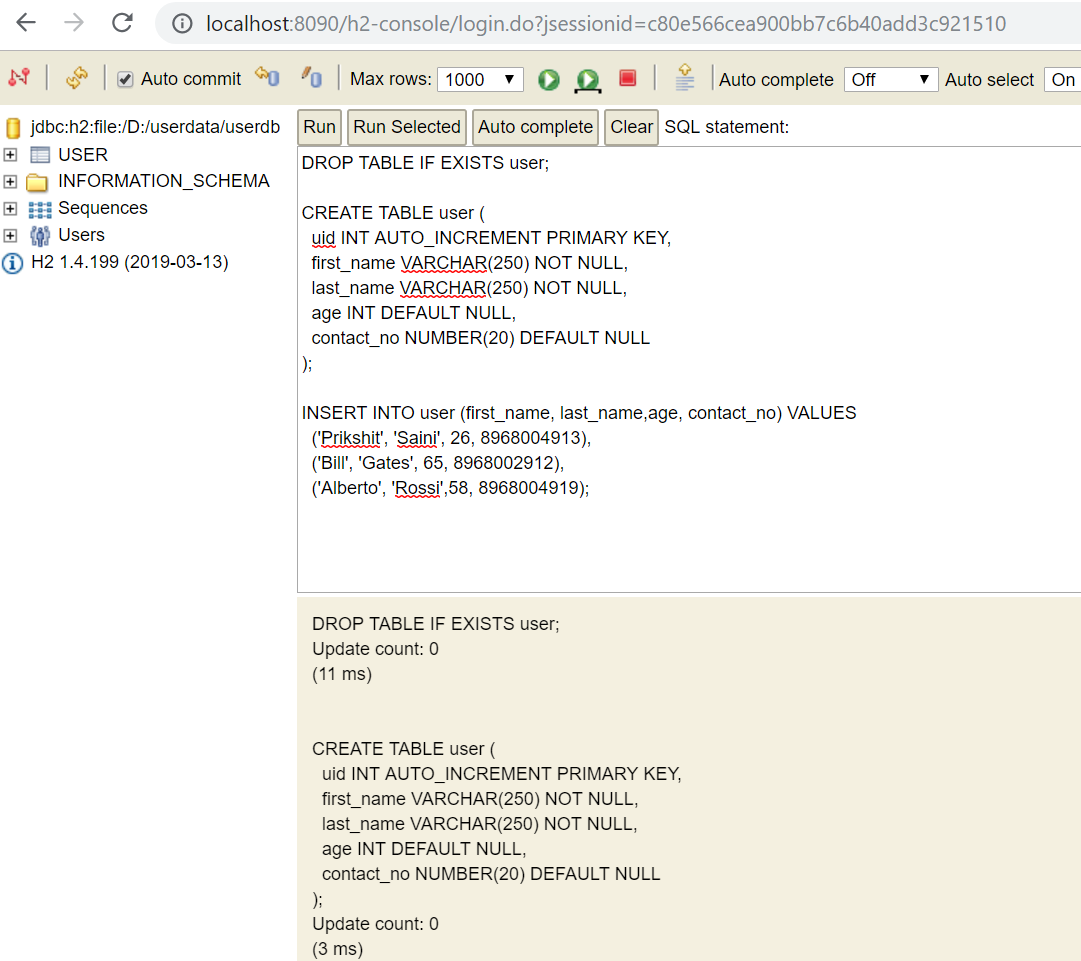


Step 4: start user-service project it will be running on port no. 8090.

Step 5: open URL <http://localhost:8090/h2-console>

Step 6: change JDBC URL to jdbc:h2:file:/D:/userdata/userdb and click on connect.

Step 7: copy script from userdb.sql file and run it against the database.

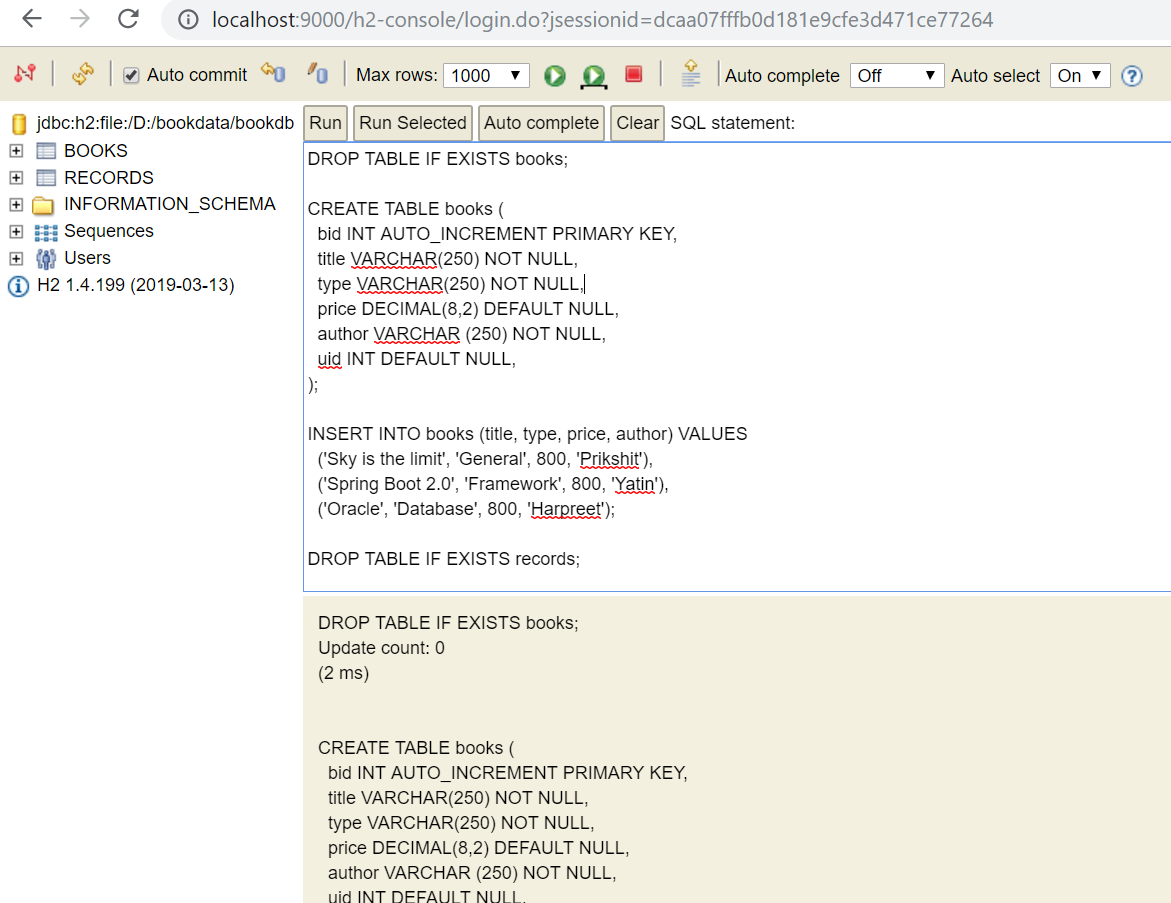


Step 8: start book-service project it will be running on port no. 9000.

Step 9: open URL <http://localhost:9000/h2-console>

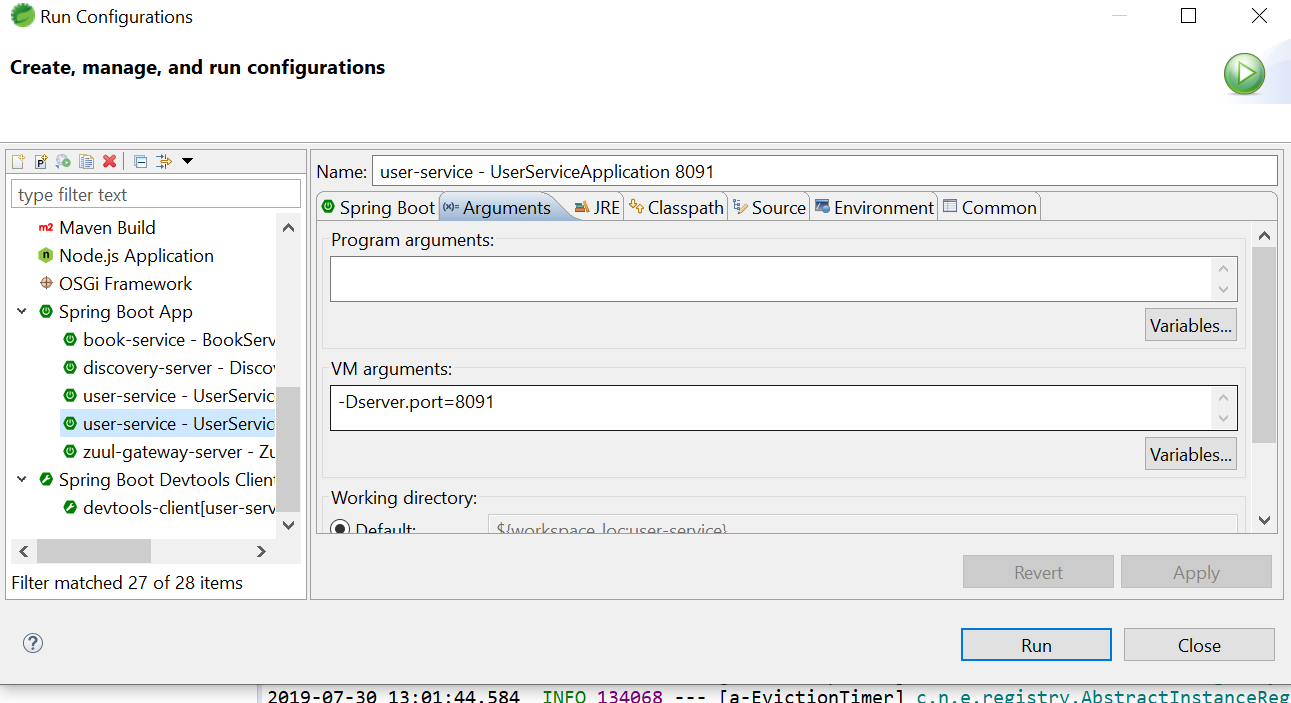
Step 10: change JDBC URL to jdbc:h2:file:/D:/bookdata/bookdb and click on connect.

Step 11: copy script from bookdb.sql file and run it against the database.



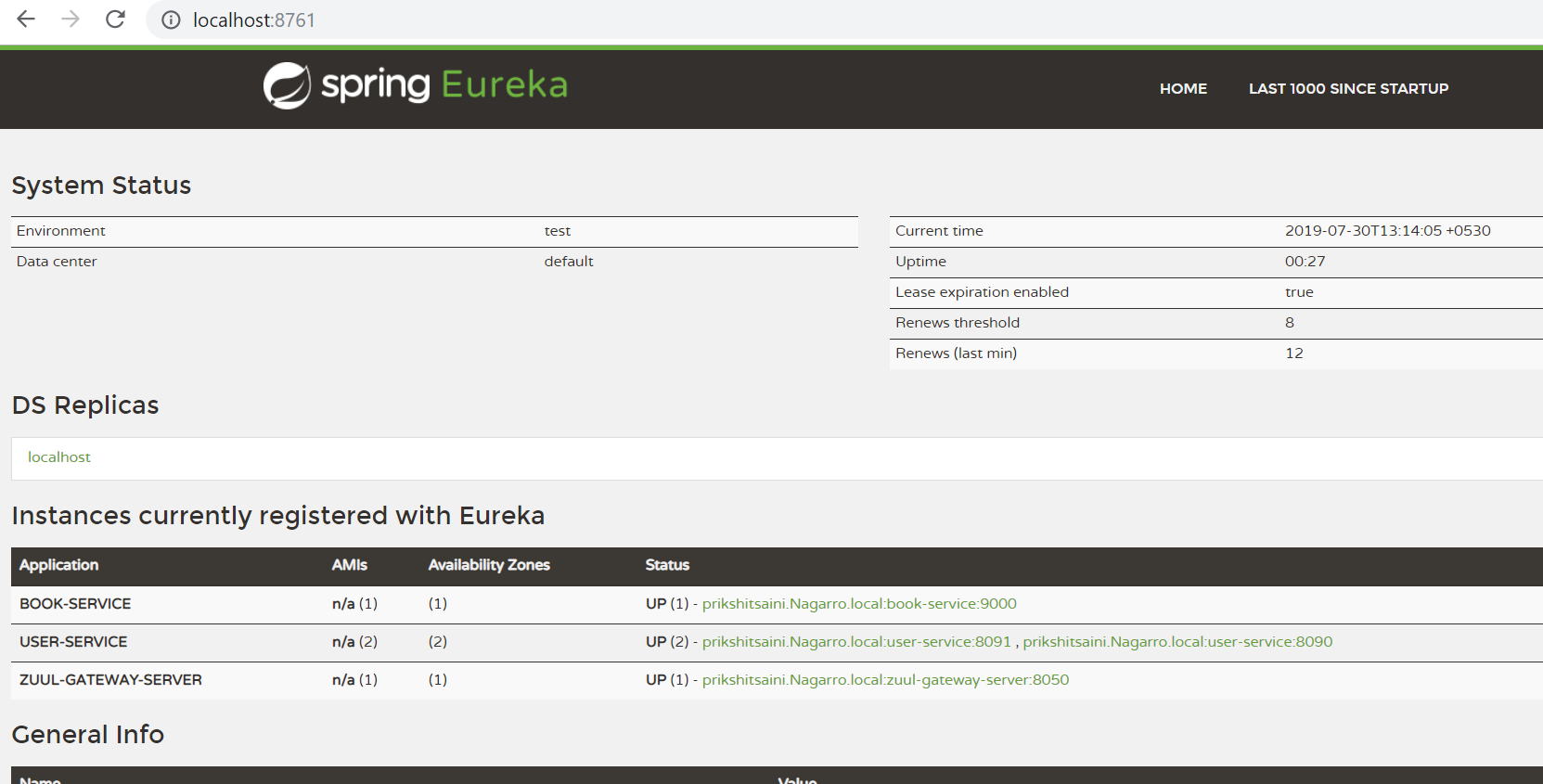
(Optional Step)

Step 13: we can run multiple instances of book and user services by changing the port number from Run configurations by passing VM arguments -Dserver.port=portnumber.



Step 14: start zuul-gateway-server project and it will be running on port no. 8050.

Step 15: open discovery server URL <http://localhost:8761> to check services are registered or not.



After all these steps everything is up and running.

**Rest Requests:**

1: Fetch user detail

http://localhost:8050/user-service/user/{userid}

GET request: <http://localhost:8050/user-service/user/1>

2: Fetch users’ detail

GET request: <http://localhost:8050/user-service/users>

3. Add new user

POST request: <http://localhost:8050/user-service/user/add>

Request body:

{

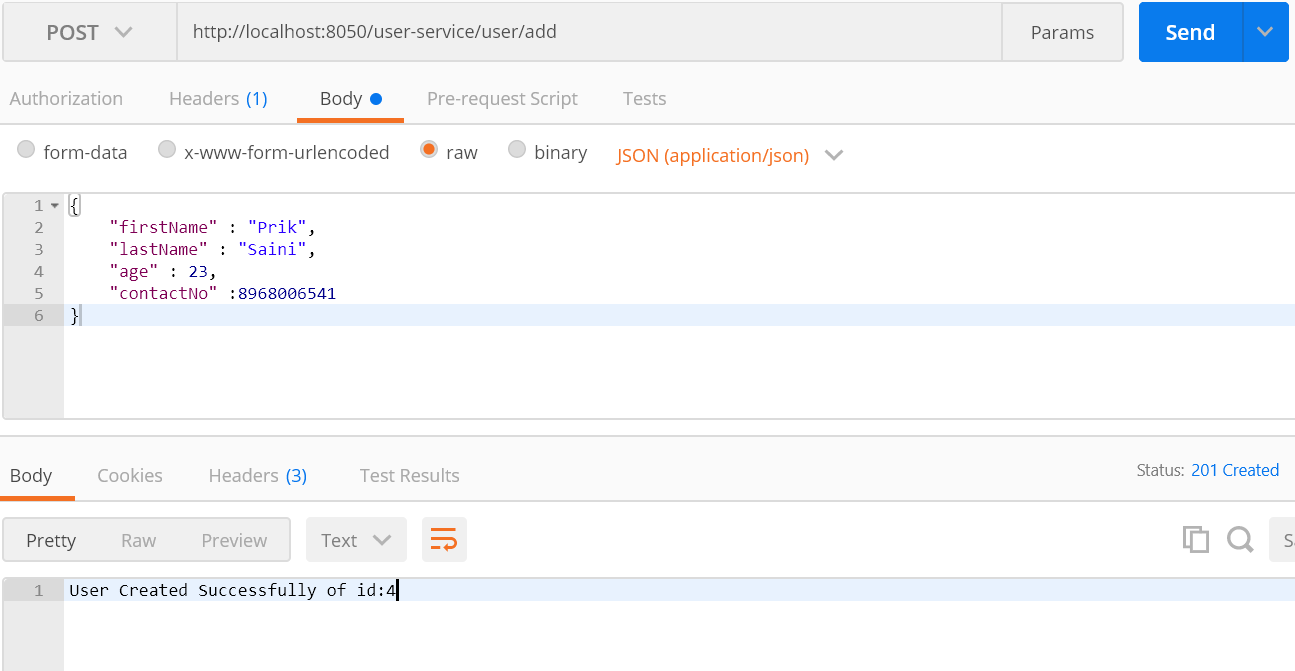
"firstName": "Prik",

"lastName": "Saini",

"age": 23,

"contactNo" :8968006541

}



4: Fetch book detail

http://localhost:8050/book-service/book/{bookid}

GET request: <http://localhost:8050/book-service/book/1>

5: Fetch books’ detail

GET request: <http://localhost:8050/book-service/books>

6. Add new book

POST request: <http://localhost:8050/book-service/book/add>

Request body:

{

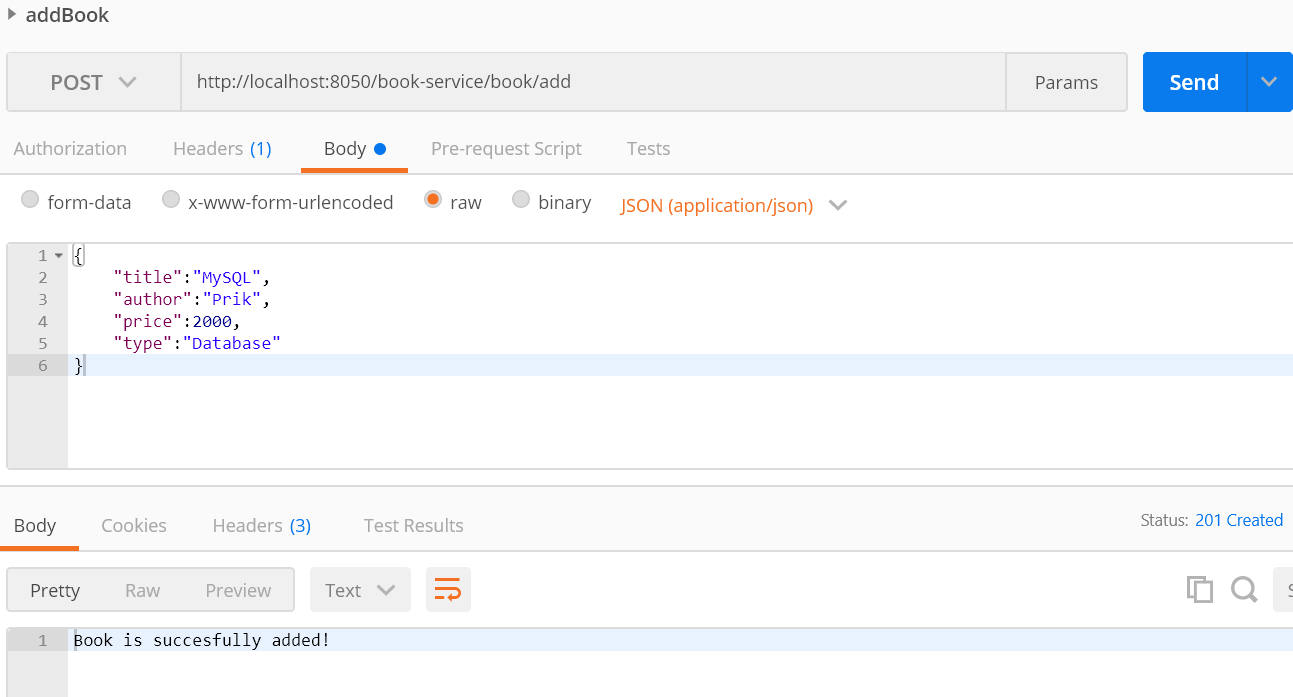
"title": "MySQL",

"author": "Prik",

"price":2000,

"type": "Database"

}



7. Issue book

http://localhost:8050/book-service/book/issue/{bookid}/user/{userid}

POST request: <http://localhost:8050/book-service/book/issue/1/user/1>

8. Similar type not allowed issue book

http://localhost:8050/book-service/book/issue/{bookid}/user/{userid}

POST request: <http://localhost:8050/book-service/book/issue/2/user/1>

9. Retrieve issued books

http://localhost:8050/book-service/books/user/{userid}

GET request: <http://localhost:8050/book-service/books/user/1>