ASSIGNMENT

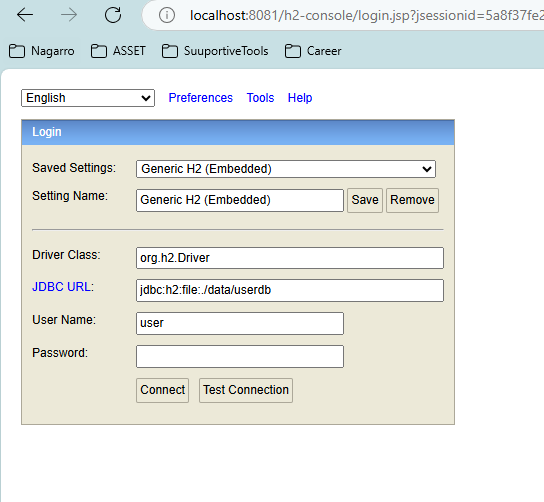
Java version 17

Steps:

1. Take a pull of the project in a codebase folder.
2. Create a new folder for workspace.
3. Import the project-> select existing maven project -> give the path of codebase folder and select all projects in it.

For user-service microservice(port is 8081): <http://localhost:8081>

1. Do right click on UserServicedApplication.java file.
2. Goto ‘Run as’ -> click on ‘Java Application’
3. When service started go to Edge/Chrome
4. Enter <http://localhost:8081/h2-console> screen appears like below image



1. Click on connect and use USERS database (refer screenshot highlighted)

A screenshot of a computer

AI-generated content may be incorrect.

1. Now go to Postman
2. (create a new user) POST <http://localhost:8081/users/new>;

Pass body {

    "name": "Fang",

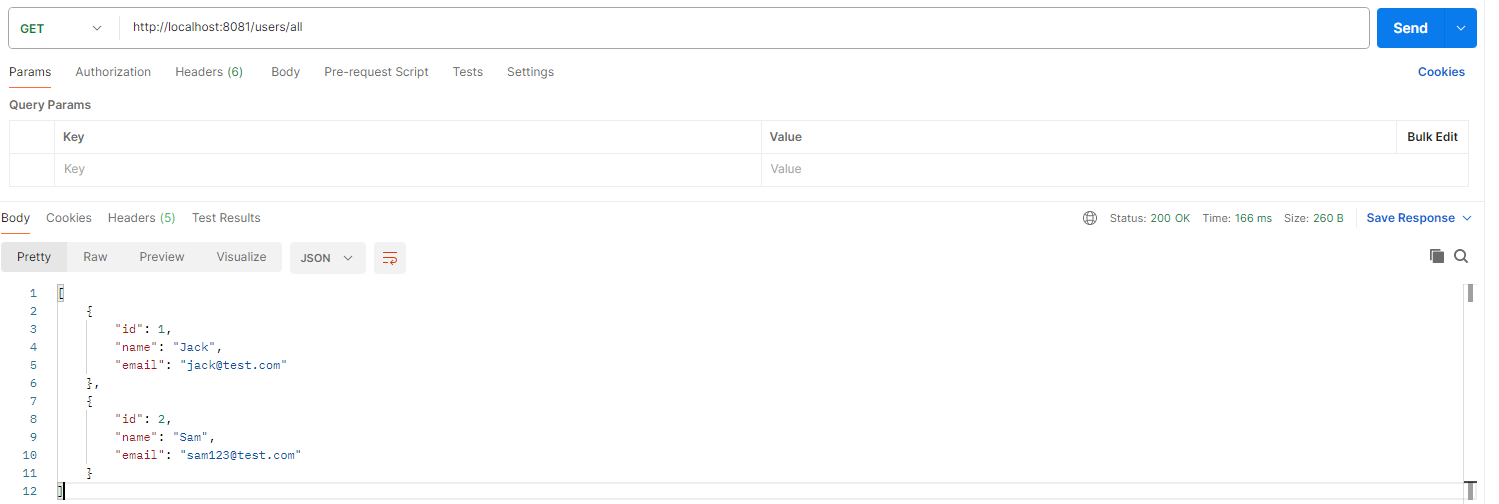
    "email": "fang@test.com"

}

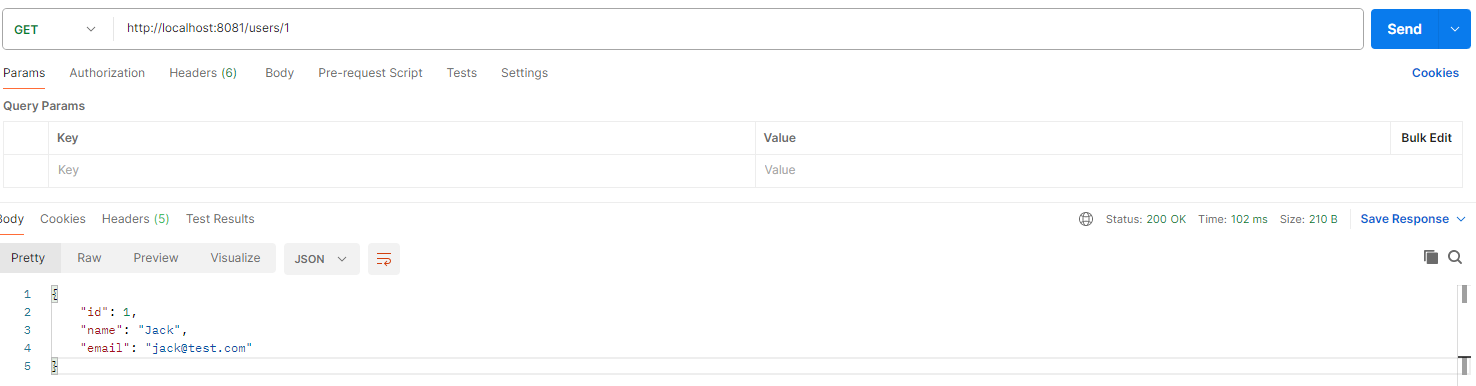
A screenshot of a computer

AI-generated content may be incorrect.

1. (get all users) GET <http://localhost:8081/users/all>



1. (get a user by userid) GET <http://localhost:8081/users/1>



1. (update the user) PUT <http://localhost:8081/users/2>

Pass body {

    "name": "Sam",

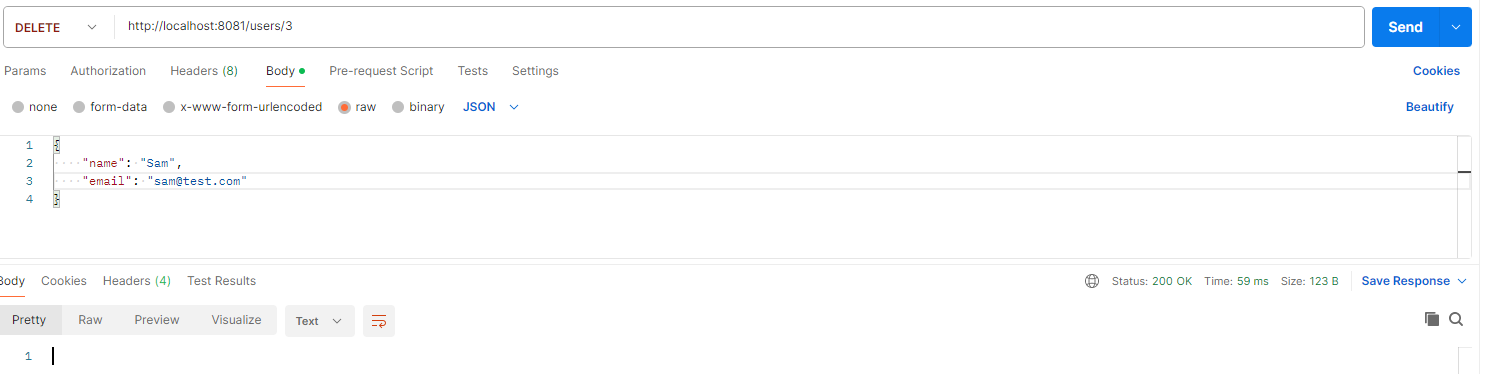
    "email": "sam123@test.com"

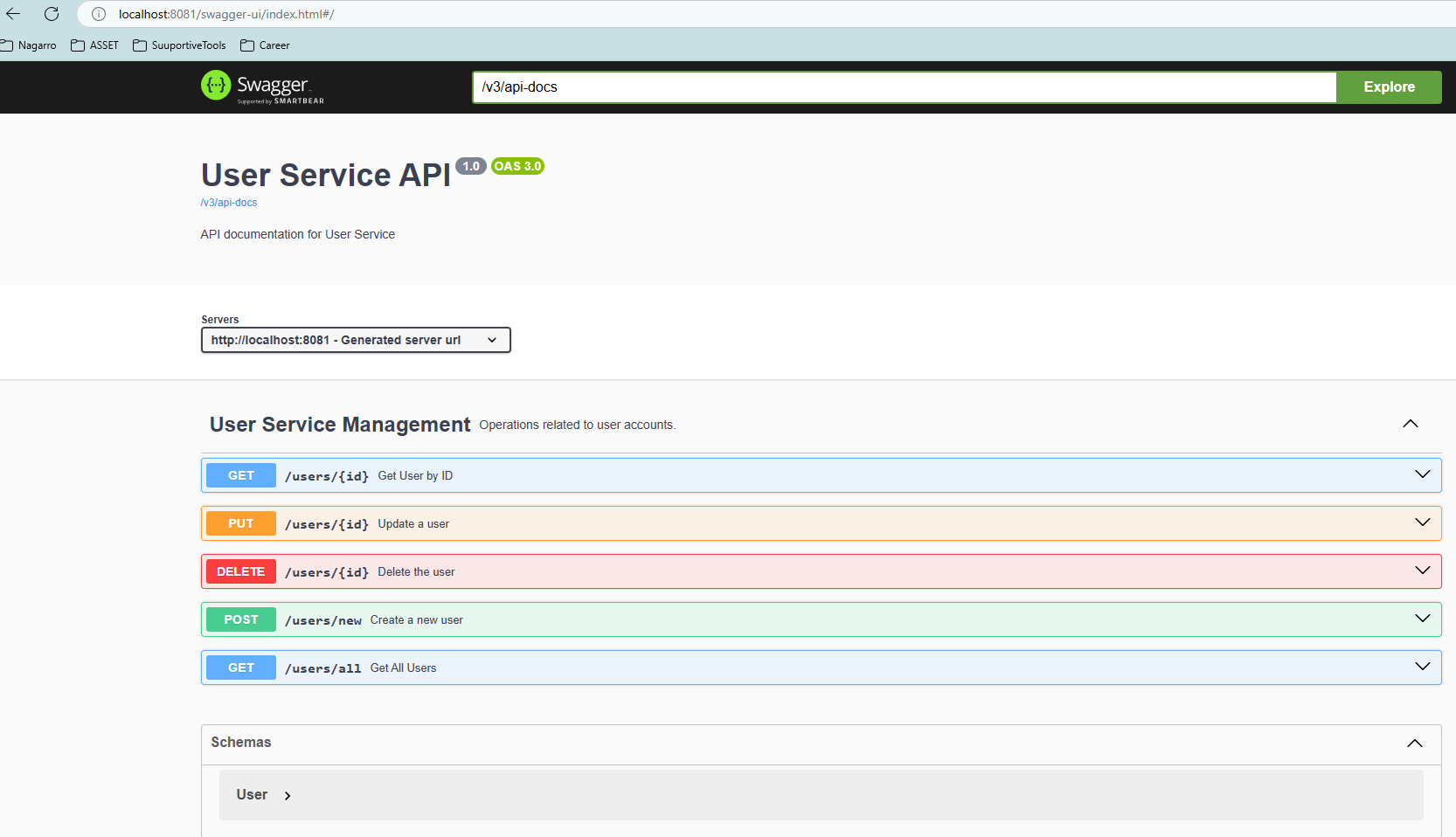
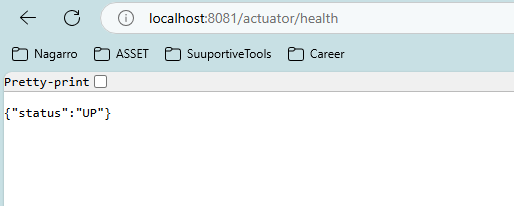
}

A screenshot of a computer

AI-generated content may be incorrect.

1. (delete the user) DELETE http://localhost:8081/users/3



1. Access the API documentation at <http://localhost:8081/swagger-ui.html>  
   
2. Access the health endpoints at <http://localhost:8081/actuator/health>  
   

For product-service microservice (port is 8082): <http://localhost:8082>

1. Do right click on ProductServiceApplication.java file.
2. Goto ‘Run as’ -> click on ‘Java Application’
3. Create database in MongoDB, with name ‘productdb’ and collection name ‘products’

A screenshot of a computer

AI-generated content may be incorrect.

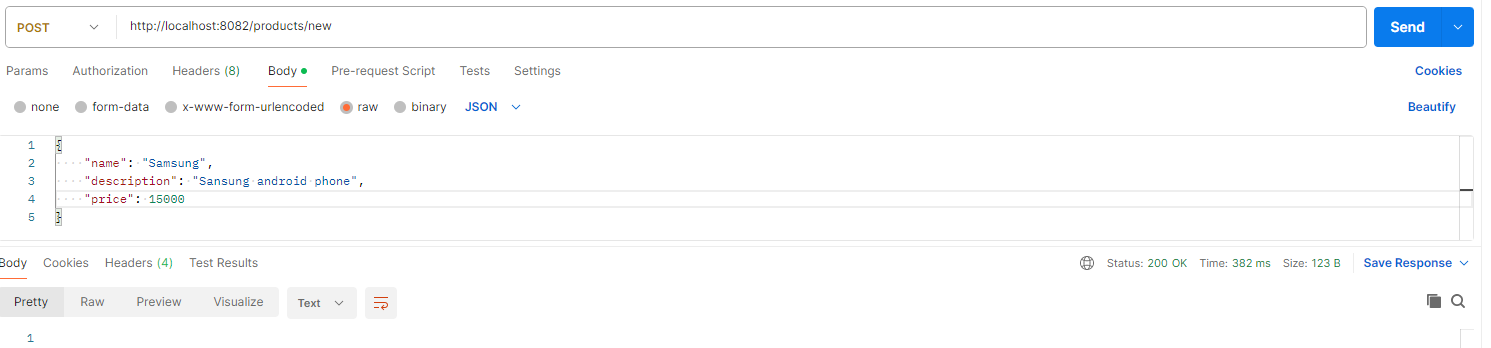
1. Now go to postman
2. (create a new product) POST <http://localhost:8082/products/new>  
   pass body {

    "name": "Samsung",

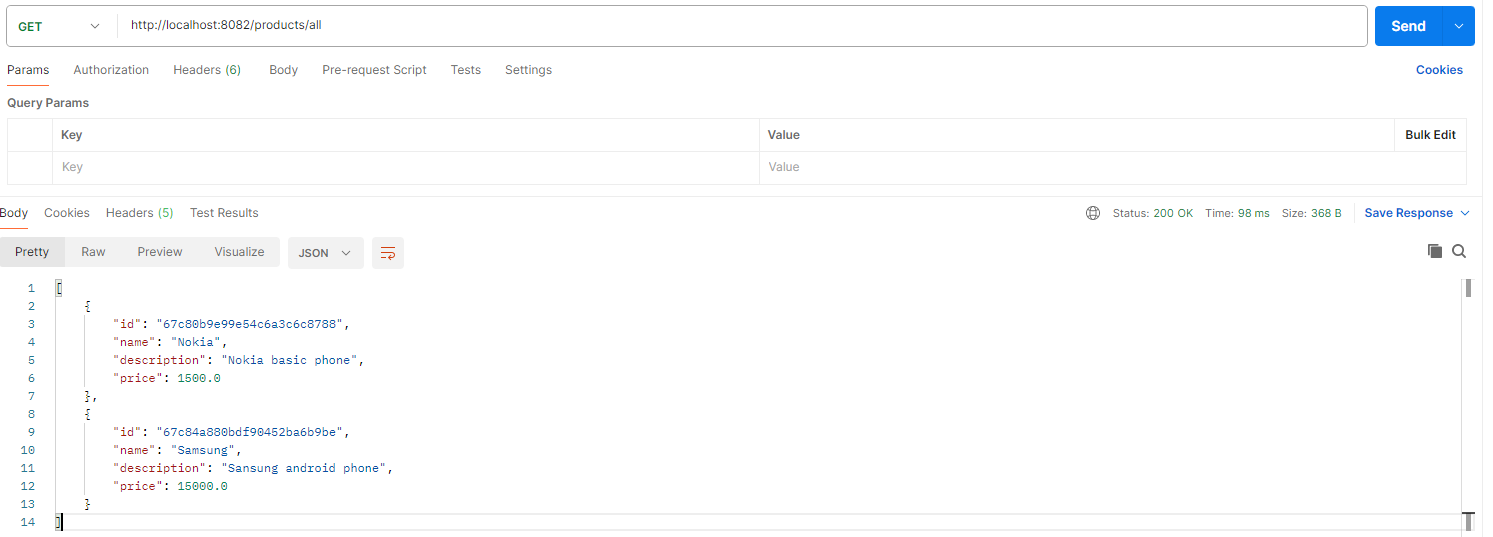
    "description": "Sansung android phone",

    "price": 15000

}



1. (get all products) GET <http://localhost:8082/products/all>



1. Access the API documentation at <http://localhost:8082/swagger-ui.html>

A computer screen shot of a computer

AI-generated content may be incorrect.

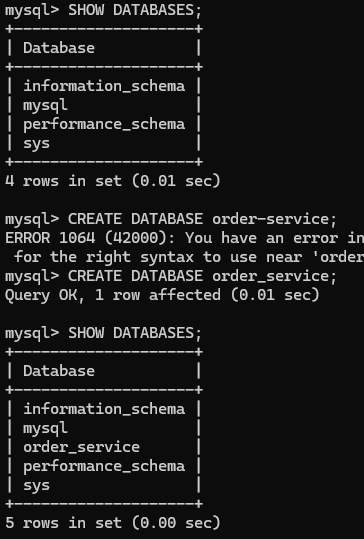
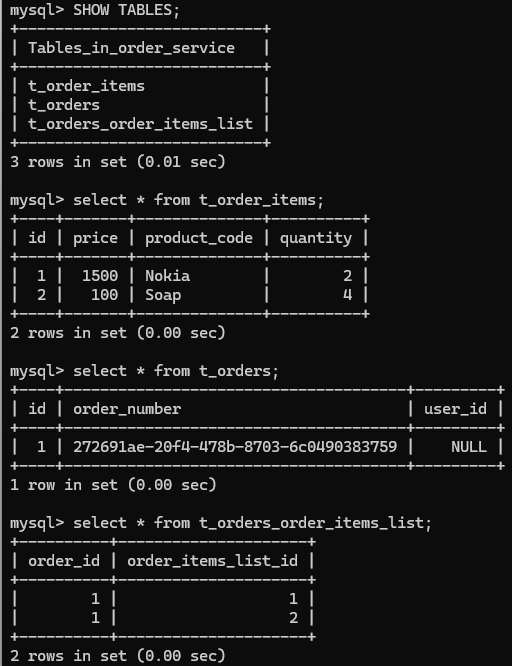
1. Access the health endpoints at [localhost:8082/actuator/health](http://localhost:8082/actuator/health)

A screenshot of a computer

AI-generated content may be incorrect.

For order-service microservice (port is 8083): <http://localhost:8083>

1. Do right click on OrderServiceApplication.java file.
2. Goto ‘Run as’ -> click on ‘Java Application’
3. Create database in MySQL, with name ‘order\_service’ and table names are ‘t\_order\_items’ and ‘t\_orders’ (for reference screenshot attached)

1. Now go to Postman

Placing the order using userId, userId is verifying by user-service microservices according to user authentication order will be placed or not.

1. (userId 2 is exists in h2 database) POST <http://localhost:8083/orders/2>   
   pass body {

    "orderItemsDtoList": [

        {

        "productCode": "Nokia",

        "price": 1500,

        "quantity": 2

        },

        {

        "productCode": "Soap",

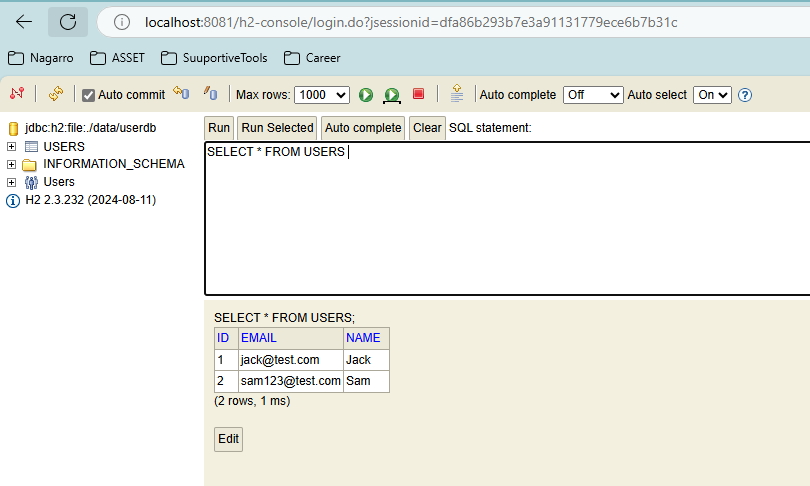
        "price": 100,

        "quantity": 4

        }

    ]

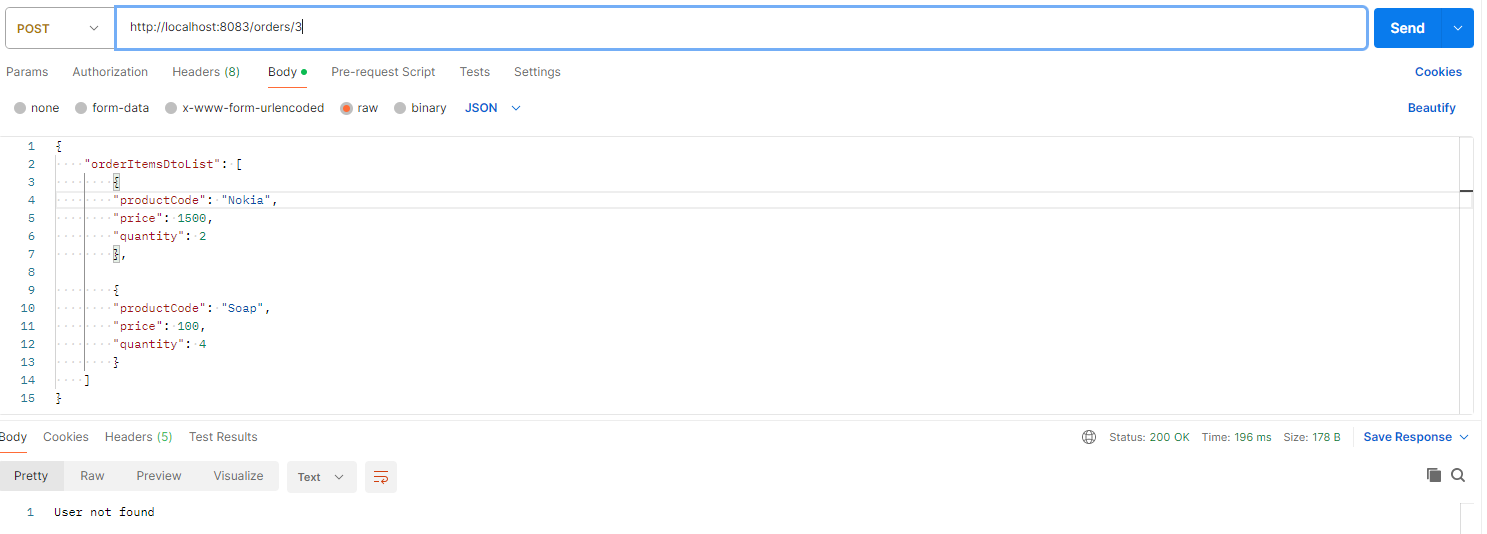
}



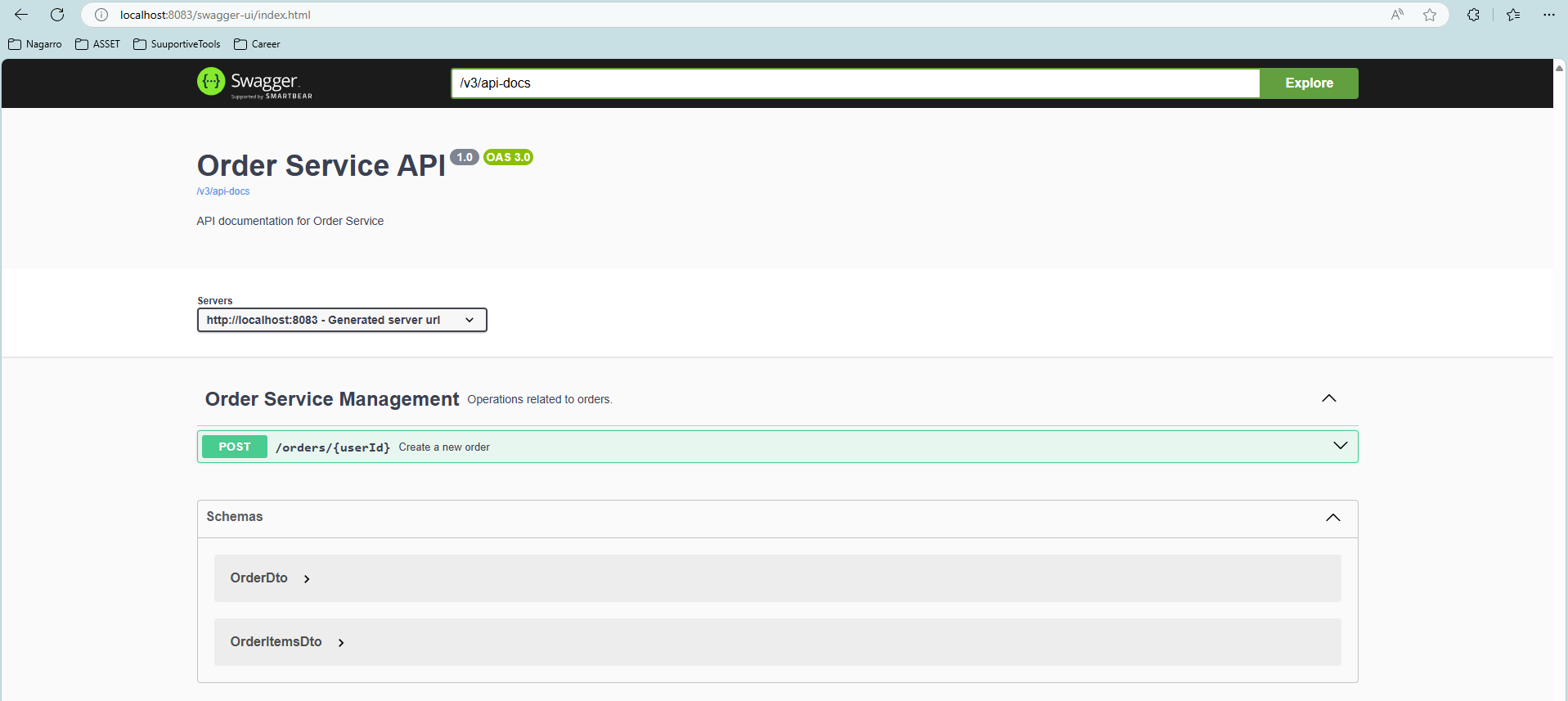
A screenshot of a computer

AI-generated content may be incorrect.

* 1. (userId 3 is not exists in h2 database) POST <http://localhost:8083/orders/3>



1. Access the API documentation at <http://localhost:8083/swagger-ui.html>



1. Access the health endpoints at [localhost:8083/actuator/health](http://localhost:8082/actuator/health)

