Adishree Devoor

Superset id : 6389680

WEEK : 05

Creating Microservices for Account and Loan

In this hands-on exercise, we create two Spring Boot microservices for a bank: one for handling accounts and one for handling loans. Each microservice will be a separate Maven-based Spring RESTful application without any backend connectivity.

# Account Microservice

Steps:

1. Create a folder with your Employee ID in D: drive.

2. Inside it, create a folder named 'microservices'.

3. Open https://start.spring.io/

4. Fill the form as follows:

Group: com.cognizant  
 Artifact: account  
 Dependencies: Spring Web, Spring Boot DevTools

5. Click Generate and download the zip file.

6. Extract 'account' folder to 'microservices'.

7. Open command prompt in the 'account' folder and run:

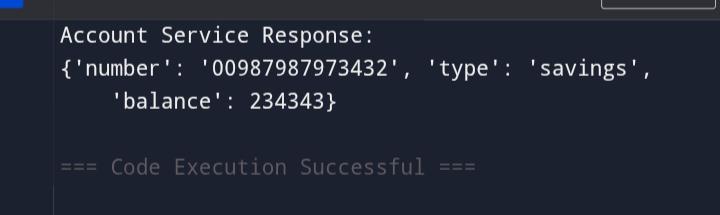
mvn clean package

8. Import project in Eclipse.

9. Create controller method as shown below:

## AccountController.java

package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
 }  
}output



# Loan Microservice

1. Follow same steps as above with the following changes:

Artifact: loan

2. Create LoanController.java with the following code:

## LoanController.java

package com.cognizant.loan.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.Map;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "car",  
 "loan", 400000,  
 "emi", 3258,  
 "tenure", 18  
 );  
 }  
}

3. Since default port 8080 is used by Account service, configure Loan service to use a different port.

4. In application.properties of Loan service, add:

server.port=8081

5. Run both services and test using browser or Postman:

http://localhost:8080/accounts/00987987973432

<http://localhost:8081/loans/H00987987972342>

Output

