Khushi

Superset id: 6390178

Week:5

Microservices Hands-On: Account and Loan Services

# Overview

This hands-on exercise demonstrates how to build two independent Spring Boot microservices – one for managing bank accounts and another for managing loans. These services do not require backend database connectivity and simply return dummy JSON responses.

# Account Microservice

1. Create a folder named with your employee ID in the D: drive.

2. Inside that folder, create another folder named 'microservices'.

3. Visit https://start.spring.io/ and fill the form as follows:

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

4. Click 'Generate' and download the zip.

5. Extract the 'account' folder into the 'microservices' directory.

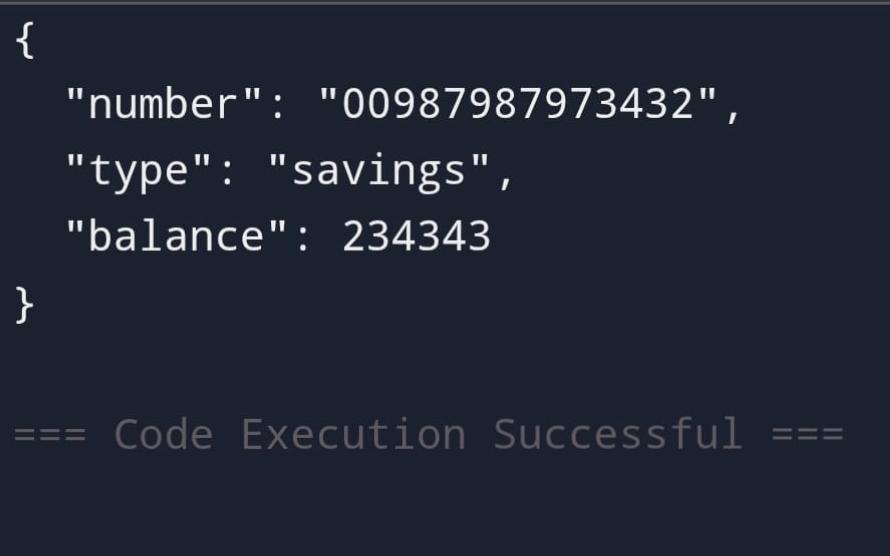
6. Open command prompt in the 'account' folder and run: mvn clean package

7. Import the project into Eclipse.

8. Create the following controller class:

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.HashMap;  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccount(@PathVariable String number) {  
 Map<String, Object> account = new HashMap<>();  
 account.put("number", number);  
 account.put("type", "savings");  
 account.put("balance", 234343);  
 return account;  
 }  
}

Output:



Loan Microservice

1. Repeat the above steps for creating another Spring Boot project with the following details:

* - Artifact: loan  
   - Group: com.cognizant  
   - Dependencies: Spring Boot DevTools, Spring Web

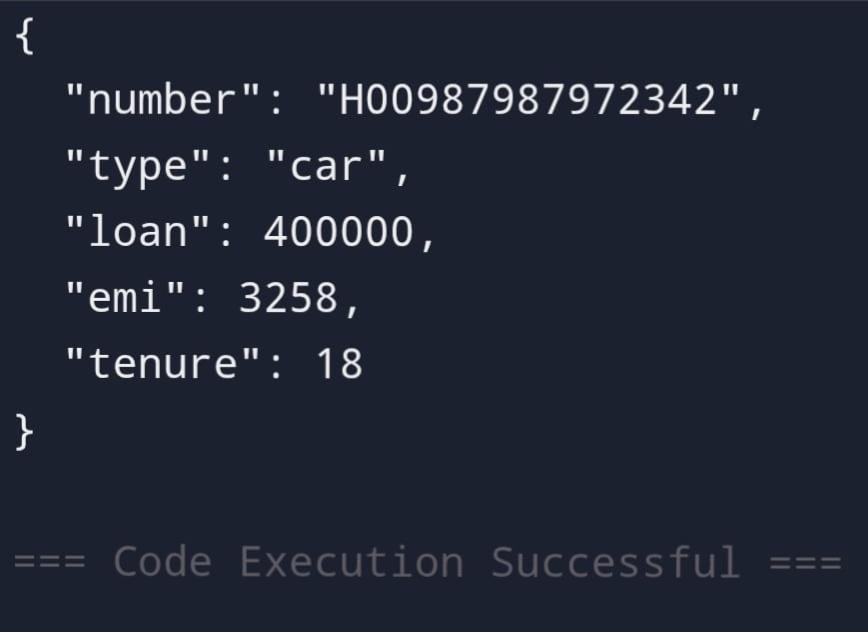
2. Extract 'loan' folder into 'microservices'.

3. Open command prompt in 'loan' folder and run: mvn clean package

4. Import into Eclipse and create the following controller:

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.HashMap;  
import java.util.Map;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoan(@PathVariable String number) {  
 Map<String, Object> loan = new HashMap<>();  
 loan.put("number", number);  
 loan.put("type", "car");  
 loan.put("loan", 400000);  
 loan.put("emi", 3258);  
 loan.put("tenure", 18);  
 return loan;  
 }  
}

Output:



# Running on Different Ports

To avoid port conflicts, change the port of the loan service by adding the following to src/main/resources/application.properties:

server.port=8081

# Testing the Services

Launch each application from Eclipse. Use the following URLs to test:

* Account Service: http://localhost:8080/accounts/00987987973432
* Loan Service: http://localhost:8081/loans/H00987987972342