Creating Microservices for account and loan Hands-on Documentation

# 1. Introduction

In this hands-on exercise, we created two independent microservices using Spring Boot:  
- Account Microservice  
- Loan Microservice  
  
Each microservice is a separate Spring Boot Maven project running on different ports and exposing a REST endpoint with dummy data, without any backend connectivity.

# 2. Folder Structure

The projects are organized in the following folder structure:  
  
D:\<EMPLOYEE\_ID>\microservices\  
 ├── account  
 └── loan

# 3. Account Microservice

Steps followed to create the Account Microservice:  
1. Generated a Spring Boot project using Spring Initializr with:  
 - Group: com.cognizant  
 - Artifact: account  
 - Dependencies: Spring Web, Spring Boot DevTools  
2. Extracted and moved it to the microservices folder.  
3. Built the project using: mvn clean package  
4. Imported into Eclipse IDE.  
5. Created a controller with the following endpoint:  
 - Method: GET  
 - URL: /accounts/{number}  
 - Response: A static dummy JSON response.

Sample Controller Code:

@GetMapping("/accounts/{number}")  
public Map<String, Object> getAccountDetails(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
}

# 4. Loan Microservice

Steps followed to create the Loan Microservice:  
1. Generated a Spring Boot project using Spring Initializr with:  
 - Group: com.cognizant  
 - Artifact: loan  
 - Dependencies: Spring Web, Spring Boot DevTools  
2. Extracted and moved it to the microservices folder.  
3. Built the project using: mvn clean package  
4. Imported into Eclipse IDE.  
5. Set a different port in application.properties:  
 - server.port=8081  
6. Created a controller with the following endpoint:  
 - Method: GET  
 - URL: /loans/{number}  
 - Response: A static dummy JSON response.

Sample Controller Code:

@GetMapping("/loans/{number}")  
public Map<String, Object> getLoanDetails(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "car",  
 "loan", 400000,  
 "emi", 3258,  
 "tenure", 18  
 );  
}

# 5. Final Outcome

Both services run independently on different ports:  
- Account Service: http://localhost:8080/accounts/{number}  
- Loan Service: http://localhost:8081/loans/{number}  
  
Each returns dummy data in JSON format. This demonstrates a basic microservices structure with Spring Boot.

# 6. Output Snapshots



