Microservices Setup: Account and Loan Services

Overview

This document describes the step-by-step process of creating two Spring Boot microservices for a bank application: Account Microservice and Loan Microservice. Each service is independent, has its own project and runs on different ports.

Prerequisites

- JDK 17 or later

- Maven installed (verify with `mvn -v`)

- Eclipse or Spring Tool Suite (STS)

- Internet connection to generate Spring Boot project

Step 1: Folder Setup

1. Navigate to the D: drive.

2. Create a folder with your employee ID, e.g., D:\123456

3. Inside that folder, create a subfolder named `microservices`.

Step 2: Create Account Microservice

1. Go to https://start.spring.io

2. Fill in the details as below:

  - Group: com.cognizant

  - Artifact: account

  - Dependencies: Spring Web, Spring Boot DevTools

3. Generate and extract the zip to D:\123456\microservices

Step 3: Build Account Project

```
cd D:\123456\microservices\account
mvn clean package
```

Step 4: Import into Eclipse

1. Open Eclipse > File > Import > Existing Maven Projects

2. Select D:\123456\microservices\account

Step 5: Add AccountController.java

```java
package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.*;
import java.util.Map;

@RestController
@RequestMapping("/accounts")
public class AccountController {

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

Step 6: Ensure AccountApplication.java

```java
package com.cognizant.account;

import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
```

```
public class AccountApplication {
  public static void main(String[] args) {
    SpringApplication.run(AccountApplication.class, args);
  }
}
```

Step 7: Run & Test

http://localhost:8080/accounts/00987987973432

Step 8: Create Loan Microservice

Repeat the above steps for Loan microservice. Use `loan` as artifact name.

```
cd D:\123456\microservices\loan
mvn clean package
```

Step 9: Add LoanController.java

```
package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.*;
import java.util.Map;

@RestController
@RequestMapping("/loans")
public class LoanController {

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

```
    }
}
```

Step 10: LoanApplication.java

```java
package com.cognizant.loan;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class LoanApplication {
  public static void main(String[] args) {
    SpringApplication.run(LoanApplication.class, args);
  }
}
```

Step 11: Change Port in application.properties

```
server.port=8081
```

Step 12: Run & Test

http://localhost:8081/loans/H00987987972342

Account Output:

Console output:

Web output:


```
{
    "type": "savings",
    "balance": 234343,
    "number": "00987987973432"
}
```

Loan Output:

Console Output:

Web output:

```
{
    "type": "car",
    "number": "H00987987972342",
    "tenure": 18,
    "emi": 3258,
    "loan": 400000
}
```

Service Summary

Service	Port	Sample Endpoint
Account Service	8080	http://localhost:8080/accounts/00987987973432
Loan Service	8081	http://localhost:8081/loans/H00987987972342