# Week 5 - Microservices with Spring Boot 3 and Spring Cloud

### **Hands-on: Creating Microservices for Account and Loan**

#### Scenario

In this hands-on exercise, we create two independent Spring Boot microservices for a bank: one for handling accounts and another for handling loans. Each microservice is a standalone Spring RESTful web service project with its own `pom.xml`, running on separate ports. The services return dummy data without any backend connectivity.

#### Steps

* Create a folder with your employee ID on D: drive.
* Inside it, create a folder named 'microservices'.
* Go to https://start.spring.io/ and generate a Spring Boot project for each microservice.
* For Account Microservice:  
   - Group: com.cognizant  
   - Artifact: account  
   - Dependencies: Spring Web, Spring Boot DevTools  
   - Endpoint: GET /accounts/{number}
* For Loan Microservice:  
   - Group: com.cognizant  
   - Artifact: loan  
   - Dependencies: Spring Web, Spring Boot DevTools  
   - Endpoint: GET /loans/{number}  
   - Add `server.port=8081` in `application.properties` to avoid port conflicts.
* Build using `mvn clean package`, import into Eclipse, run and test using a browser.

#### Account Microservice Code

**File Name: account/src/main/java/com/cognizant/account/AccountApplication.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);  
 }  
}

**File Name: account/src/main/java/com/cognizant/account/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  
 );  
 }  
}

**File Name: account/pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.cognizant</groupId>  
 <artifactId>account</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.0</version>  
 <relativePath/>  
 </parent>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

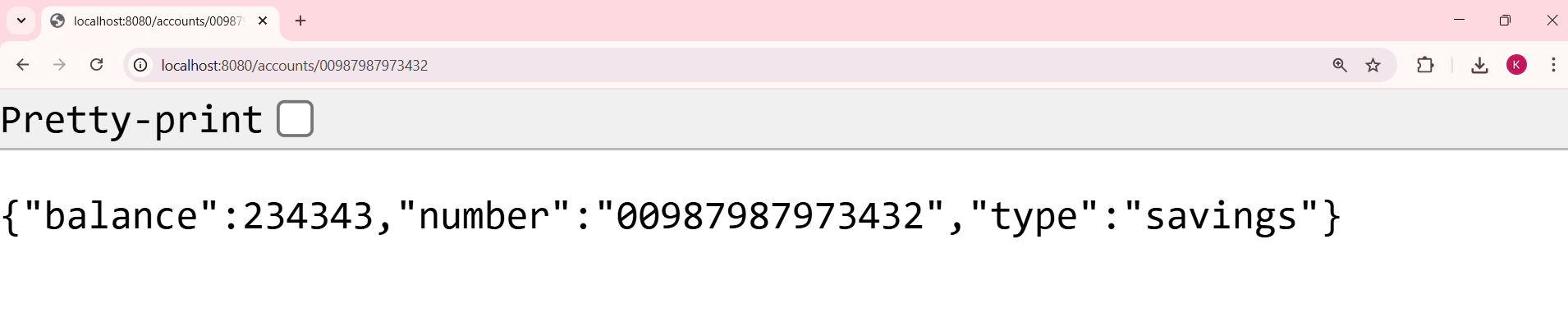
**File Name:account/src/main/resources/application.properties**

spring.application.name=account-service

# Runs on default port 8080, no extra configuration needed

#### Sample Output

Account Microservice (GET http://localhost:8080/accounts/00987987973432):



#### Loan Microservice Code

**File Name: loan/src/main/java/com/cognizant/loan/LoanApplication.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);  
 }  
}

**File Nmae: loan/src/main/java/com/cognizant/loan/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  
 );  
 }  
}

**File Name:loan/src/main/resources/application.properties**

server.port=8081

**File Nmae: pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0   
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.cognizant</groupId>  
 <artifactId>loan</artifactId>  
 <version>0.0.1-SNAPSHOT</version>  
 <packaging>jar</packaging>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.0</version>  
 <relativePath/>  
 </parent>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
</project>

#### Sample Output

Loan Microservice (GET http://localhost:8081/loans/H00987987972342):

