**WEEK 5**

**Creating Microservices for account and loan**

In this hands on exercises, we will create two microservices for a bank. One microservice for handing accounts and one for handling loans. Each microservice will be a specific independent Spring RESTful Webservice maven project having it's own pom.xml. The only difference is that, instead of having both account and loan as a single application, it is split into two different applications. These webservices will be a simple service without any backend connectivity.

**1. 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);

}

}

**2. Loan.java (Model)**

package com.cognizant.loan.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Loan {

@Id

private int id;

private String name;

private double amount;

public Loan() {

}

public Loan(int id, String name, double amount) {

this.id = id;

this.name = name;

this.amount = amount;

}

// Getters and Setters

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getAmount() {

return amount;

}

public void setAmount(double amount) {

this.amount = amount;

}

}

**3. LoanRepository.java**

package com.cognizant.loan.repository;

import com.cognizant.loan.model.Loan;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface LoanRepository extends JpaRepository<Loan, Integer> {

}

**4. LoanController.java**

package com.cognizant.loan.controller;

import com.cognizant.loan.model.Loan;

import com.cognizant.loan.repository.LoanRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

@RestController

@RequestMapping("/loans")

public class LoanController {

@Autowired

private LoanRepository loanRepository;

@GetMapping("/{number}")

public Loan getLoanById(@PathVariable int number) {

return loanRepository.findById(number).orElse(null);

}

}

**5. application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/your\_database\_name

spring.datasource.username=your\_username

spring.datasource.password=your\_password

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**pom.xml (Spring Boot + JPA + Web + MySQL)**

xml

CopyEdit

<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.loan</groupId>

<artifactId>loan</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Loan Service</name>

<description>Loan microservice using Spring Boot</description>

<packaging>jar</packaging>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version> <!-- You can use the latest version -->

<relativePath/> <!-- lookup parent from repository -->

</parent>

<properties>

<java.version>17</java.version> <!-- Your Eclipse is using Java 17 -->

</properties>

<dependencies>

<!-- Spring Boot Web Starter -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<!-- Spring Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- MySQL Driver -->

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<!-- Spring Boot Test -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<!-- Maven Plugin for Spring Boot -->

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>${spring-boot.version}</version>

</plugin>

</plugins>

</build>

</project>

**OUTPUT**