**CREATING MICROSERVICES FOR ACCOUNT AND LOAN**

**( WEEK 5)**

**SUPER SET ID:6410152**

Account-service Microservice:

**Step 1: Create Spring Boot Project in Eclipse**

1. Open **Eclipse**.
2. Go to **File → New → Spring Starter Project**.
3. Fill in:
   * **Project Name**: account-service
   * **Group**: com.example
   * **Artifact**: account
   * **Type**: Maven
   * **Packaging**: Jar
4. Click **Finish**.

Account class:

package com.example.account.model;

import jakarta.persistence.\*;

@Entity

public class Account {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String accountNumber;

private String accountHolderName;

private Double balance;

// Getters and Setters

}

**Repository: AccountRepository.java**

package com.example.account.repository;

import com.example.account.model.Account;

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

public interface AccountRepository extends JpaRepository<Account, Long> {}

**Controller: AccountController.java :**

package com.example.account.controller;

import com.example.account.model.Account;

import com.example.account.repository.AccountRepository;

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

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

import java.util.List;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@Autowired

private AccountRepository accountRepo;

@PostMapping

public Account create(@RequestBody Account account) {

return accountRepo.save(account);

}

@GetMapping

public List<Account> getAll() {

return accountRepo.findAll();

}

@GetMapping("/{id}")

public Account getById(@PathVariable Long id) {

return accountRepo.findById(id).orElse(null);

}

}

**Application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/accountdb

spring.datasource.username=root

spring.datasource.password=your\_password

spring.jpa.hibernate.ddl-auto=update

server.port=8081

📌 Use H2 instead of MySQL if you want to avoid DB setup for now:

spring.datasource.url=jdbc:h2:mem:testdb

spring.h2.console.enabled=true

**Run the Application:**

1. Right-click AccountServiceApplication.java
2. Click **Run As > Java Application**

Visit: <http://localhost:8081/accounts>

Creating loan-service:

**Add Code for Loan Microservice**

**Loan.java:**

package com.example.loan.model;

import jakarta.persistence.\*;

@Entity

public class Loan {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String loanNumber;

private Double amount;

private String status;

// Getters and Setters

}

**Repository: LoanRepository.java**

package com.example.loan.repository;

import com.example.loan.model.Loan;

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

public interface LoanRepository extends JpaRepository<Loan, Long> {}

**Controller: LoanController.java**

package com.example.loan.controller;

import com.example.loan.model.Loan;

import com.example.loan.repository.LoanRepository;

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

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

import java.util.List;

@RestController

@RequestMapping("/loans")

public class LoanController {

@Autowired

private LoanRepository loanRepo;

@PostMapping

public Loan create(@RequestBody Loan loan) {

return loanRepo.save(loan);

}

@GetMapping

public List<Loan> getAll() {

return loanRepo.findAll();

}

@GetMapping("/{id}")

public Loan getById(@PathVariable Long id) {

return loanRepo.findById(id).orElse(null);

}

}

**Application.properties:**

**For MySQL:**

spring.datasource.url=jdbc:mysql://localhost:3306/loandb

spring.datasource.username=root

spring.datasource.password=your\_password

spring.jpa.hibernate.ddl-auto=update

server.port=8082

**For H2 (quick testing):**

spring.datasource.url=jdbc:h2:mem:loandb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.h2.console.enabled=true

server.port=8082

**Run the Loan Service**

1. Right-click LoanServiceApplication.java
2. Choose **Run As > Java Application**
3. Visit: <http://localhost:8082/loans>