**Week-5**

**Microservices**

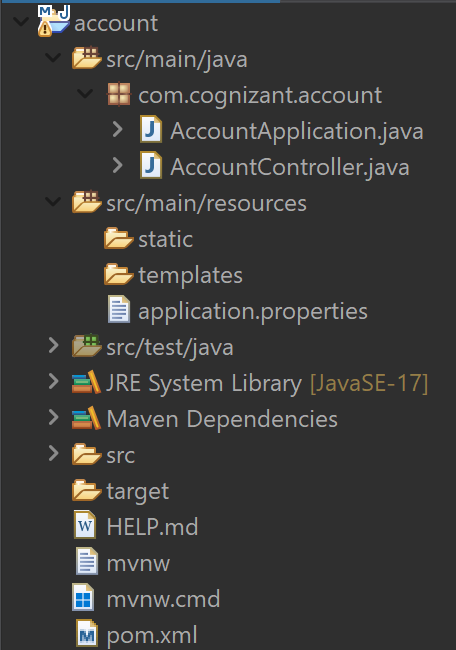
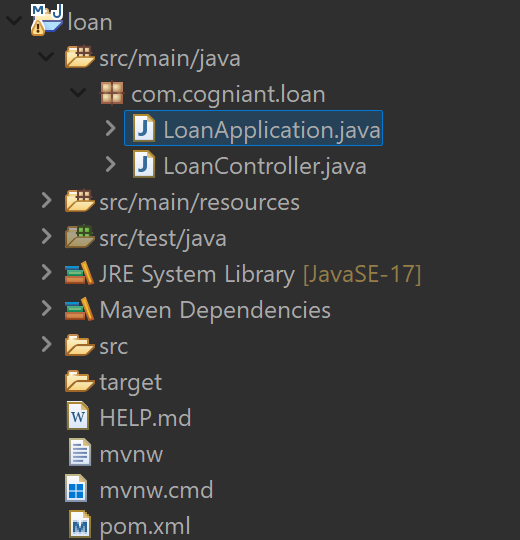
**Exercise 1: Creating Microservices for account and loan**

**Steps:**

1. **Set Up a Maven Projects:**
   * Create two separate Maven project named account and loan.
   * Import both the extracted folders into eclipse.
2. **Create the REST Controllers:**
   * Create in both the projects the REST Controllers named AccountController and LoanController respectively.
3. **Configure the Server ports:**
   * Configure the service to run on different ports.
4. **Run the Application:**
   * Create the AccountApplication and LoanApplication classes visit the respective url’s and verify the execution.

Microservices is an architectural approach where an application is built as a collection of small, independent services, each handling a specific business function. It improves scalability, flexibility, and allows teams to develop, deploy, and maintain services independently.

**Folder Structure:**

****

**LoanApplication.java**

// LoanApplication.java

package com.cogniant.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);

}

}

**AccountApplication.java**

// AccountApplication.java

package com.cogniant.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);

}

}

**LoanController.java**

// LoanController.java

package com.cogniant.loan;

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", "home",

"loan", 800000,

"emi", 5000,

"tenure", 14

);

}

}

**AccountController.java**

// AccountController.java

package com.cognizant.account;

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",

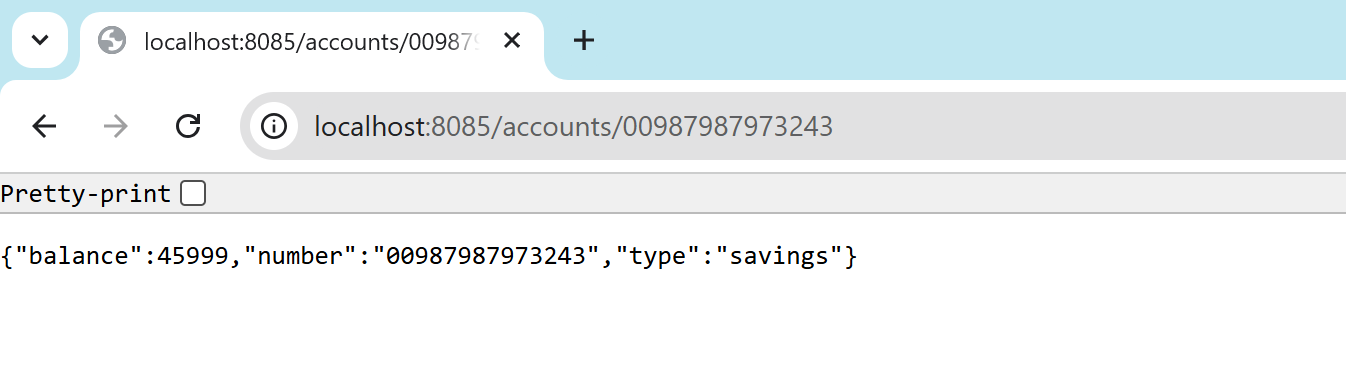
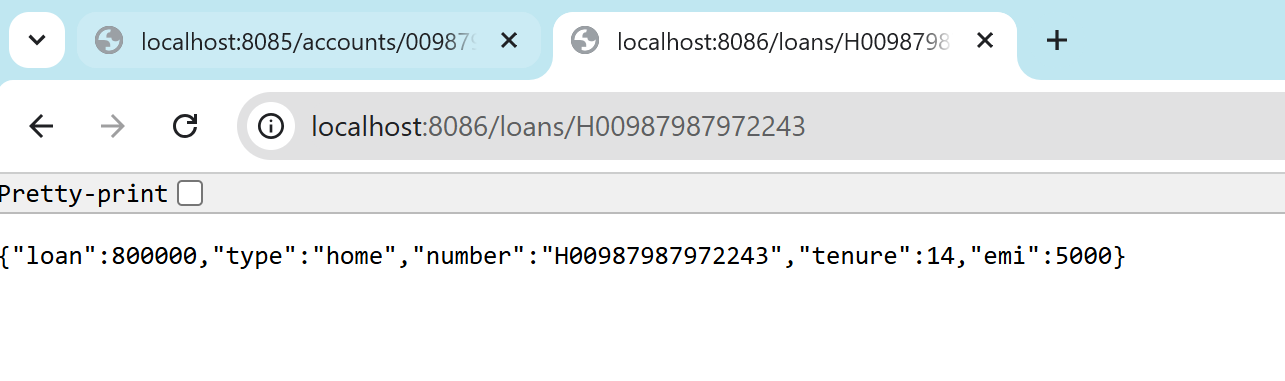
"balance", 45999

);

}

}

**Output:**

**Conclusion:**

In conclusion, microservices offer a modular and efficient way to build modern applications by breaking them into smaller, manageable components. This approach enhances scalability, simplifies maintenance, and enables independent development and deployment. It is especially useful in large, evolving systems where agility and reliability are critical.

**-- THE END --**