**CREATING MICROSERVICES FOR ACCOUNT AND LOAN**

CONTEXT OF THE EXAMPLE:

In this exercise, we create two independent Spring Boot microservices: one for managing Account details and another for Loan information. Each service runs on a separate port and exposes REST APIs to fetch respective data. This demonstrates the decentralized architecture of microservices, enabling modular development, deployment, and independent scaling.

CODE:

**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);  
 }  
  
}

**AccountController.java:**

package com.cognizant.account.controller;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/accounts")  
public class AccountController {  
  
 @GetMapping("/{number}")  
 public Account getAccount(@PathVariable String number) {  
 return new Account(number, "savings", 234343);  
 }  
 static class Account {  
 public String number;  
 public String type;  
 public double balance;  
 public Account(String number, String type, double balance) {  
 this.number = number;  
 this.type = type;  
 this.balance = balance;  
 }  
 }  
}

**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);  
 }  
  
}

**LoanController.java:**

package com.cognizant.loan.controller;  
import org.springframework.web.bind.annotation.\*;  
@RestController  
@RequestMapping("/loans")  
public class LoanController {  
  
 @GetMapping("/{number}")  
 public Loan getLoan(@PathVariable String number) {  
 return new Loan(number, "car", 400000, 3258, 18);  
 }  
 static class Loan {  
 public String number;  
 public String type;  
 public double loan;  
 public int emi;  
 public int tenure;  
  
 public Loan(String number, String type, double loan, int emi, int tenure) {  
 this.number = number;  
 this.type = type;  
 this.loan = loan;  
 this.emi = emi;  
 this.tenure = tenure;  
 }  
 }  
}

**Application.properties:**

spring.application.name=loan

server.port=8081

**BankController.java:**

package com.cognizant.bankfrontend.controller;  
import com.cognizant.bankfrontend.model.Account;  
import com.cognizant.bankfrontend.model.Loan;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.\*;  
import org.springframework.web.client.RestTemplate;  
  
@Controller  
public class BankController {  
 @Autowired  
 private RestTemplate restTemplate;  
 @GetMapping("/")  
 public String showForm() {  
 return "search"; // Loads search.html from templates  
 }  
 @PostMapping("/details")  
 public String getDetails(@RequestParam String accountNumber, Model model) {  
 // Call Account Service  
 Account account = restTemplate.getForObject(  
 "http://localhost:8080/accounts/" + accountNumber, Account.class);  
  
 // Call Loan Service  
 Loan loan = restTemplate.getForObject(  
 "http://localhost:8081/loans/" + accountNumber, Loan.class);  
  
 // Add responses to model  
 model.addAttribute("account", account);  
 model.addAttribute("loan", loan);  
  
 return "details"; // Loads details.html from templates  
 }  
}

**Account.java:**

package com.cognizant.bankfrontend.model;  
  
public class Account {  
 private String number;  
 private String type;  
 private double balance;  
  
 // Getters & Setters  
 public String getNumber() { return number; }  
 public void setNumber(String number) { this.number = number; }  
  
 public String getType() { return type; }  
 public void setType(String type) { this.type = type; }  
  
 public double getBalance() { return balance; }  
 public void setBalance(double balance) { this.balance = balance; }  
}

**Loan.java:**

package com.cognizant.bankfrontend.model;  
  
public class Loan {  
 private String number;  
 private String type;  
 private double loan;  
 private int emi;  
 private int tenure;  
  
 // Getters & Setters  
 public String getNumber() { return number; }  
 public void setNumber(String number) { this.number = number; }  
  
 public String getType() { return type; }  
 public void setType(String type) { this.type = type; }  
  
 public double getLoan() { return loan; }  
 public void setLoan(double loan) { this.loan = loan; }  
  
 public int getEmi() { return emi; }  
 public void setEmi(int emi) { this.emi = emi; }  
  
 public int getTenure() { return tenure; }  
 public void setTenure(int tenure) { this.tenure = tenure; }  
}

**BankFrontendApplication.java:**

package com.cognizant.bankfrontend;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.boot.web.client.RestTemplateBuilder;  
import org.springframework.context.annotation.Bean;  
import org.springframework.web.client.RestTemplate;  
  
@SpringBootApplication  
public class BankFrontendApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(BankFrontendApplication.class, args);  
 }  
  
 @Bean  
 public RestTemplate restTemplate(RestTemplateBuilder builder) {  
 return builder.build(); // <-- This is the part you may have missed earlier  
 }  
}

**Search.html:**

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Bank Account Search</title>

</head>

<body>

<h2>Enter Account Number</h2>

<form action="/details" method="post">

<input type="text" name="accountNumber" placeholder="Account number" required>

<button type="submit">Get Details</button>

</form>

</body>

</html>

Details.html:

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Account and Loan Details</title>

</head>

<body>

<h2>Account Details</h2>

<p><strong>Number:</strong> <span th:text="${account.number}"></span></p>

<p><strong>Type:</strong> <span th:text="${account.type}"></span></p>

<p><strong>Balance:</strong> ₹<span th:text="${account.balance}"></span></p>

<h2>Loan Details</h2>

<p><strong>Number:</strong> <span th:text="${loan.number}"></span></p>

<p><strong>Type:</strong> <span th:text="${loan.type}"></span></p>

<p><strong>Loan Amount:</strong> ₹<span th:text="${loan.loan}"></span></p>

<p><strong>EMI:</strong> ₹<span th:text="${loan.emi}"></span></p>

<p><strong>Tenure:</strong> <span th:text="${loan.tenure}"></span> months</p>

</body>

</html>

Application.properties:

spring.application.name=bank-frontend

server.port=8082

**OUTPUTS:**











