**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.

**Solution:**

**Account Microservice**

**Project Structure**

account/

├── src/

│ └── main/

│ ├── java/

│ │ └── com/

│ │ └── cognizant/

│ │ └── account/

│ │ ├── AccountApplication.java

│ │ └── controller/

│ │ └── AccountController.java

│ └── resources/

│ └── application.properties

│ ── build.gradle / pom.xml

**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 getAccountDetails(@PathVariable String number) {

return new Account(number, "savings", 234343);

}

static class Account {

private String number;

private String type;

private double balance;

public Account(String number, String type, double balance) {

this.number = number;

this.type = type;

this.balance = balance;

}

public String getNumber() { return number; }

public String getType() { return type; }

public double getBalance() { return balance; }

}

}

**application.properties**

server.port=5050

**Endpoint Test**

<http://localhost:5050/accounts/00987987973432>

**Sample Response:**

{

"number": "00987987973432",

"type": "savings",

"balance": 234343

}

**Output Screenshot**

A screenshot of a computer

AI-generated content may be incorrect.

**Loan Microservice**

**Project Structure**

loan/

├── src/

│ └── main/

│ ├── java/

│ │ └── com/

│ │ └── cognizant/

│ │ └── loan/

│ │ ├── LoanApplication.java

│ │ └── controller/

│ │ └── LoanController.java

│ └── resources/

│ └── application.properties

├── build.gradle / pom.xml

**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 getLoanDetails(@PathVariable String number) {

return new Loan(number, "car", 400000, 3258, 18);

}

static class Loan {

private String number;

private String type;

private double loan;

private double emi;

private int tenure;

public Loan(String number, String type, double loan, double emi, int tenure) {

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

public String getNumber() { return number; }

public String getType() { return type; }

public double getLoan() { return loan; }

public double getEmi() { return emi; }

public int getTenure() { return tenure; }

}

}

**application.properties**

server.port=5050

**Endpoint Test**

<http://localhost:8081/loans/H00987987972342>

**Sample Response:**

{

"number": "H00987987972342",

"type": "car",

"loan": 400000,

"emi": 3258,

"tenure": 18

}

**Output Screenshot:**

**A screenshot of a computer

AI-generated content may be incorrect.**