**Microservices with Spring Boot 3 and Spring Cloud**

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

**Source Code**

**ACCOUNT MICROSERVICE**

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;

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", 234343.00);

}

record Account(String number, String type, double balance) {}

}

Application.properties

server.port=8080

pom.xml

<dependencies>

<dependency>

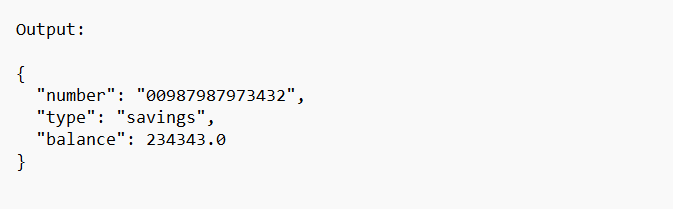
<groupId>org.springframework.boot</groupId>

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

</dependency>

</dependencies>

**OUTPUT**



**LOAN MICROSERVICE**

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;

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", 400000.0, 3258.0, 18);

}

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

}

Application.properties

server.port=8081

pom.xml

<dependencies>

<dependency>

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

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

</dependency>

</dependencies>

**OUTPUT**

