**Microservices**

**Hands On - Creating Microservices for account and loan**

**Account Model – Account.java**

**Code:**

package com.cognizant.account.Model;

public class Account{

private String number;

private int balance;

private String type;

public Account(String number, String type, int balance)

{

this.balance = balance;

this.number=number;

this.type=type;

}

public String getNumber()

{

return number;

}

public int getBalance()

{

return balance;

}

public String getType()

{

return type;

}

}

**Account Controller – AccountController.java**

**Code:**

package com.cognizant.account.controller;

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

import com.cognizant.account.Model.\*;

*@RestController*

*@RequestMapping*("/accounts")

public class AccountController{

*@GetMapping*("/{number}")

public Account getAccount(*@PathVariable* String number)

{

return new Account(number, "savings", 23434);

}

}

**AccountApplication.java running on port 7000**

**Code:**

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);

}

}

**Loan Model – Loan.java**

**Code:**

package com.cognizant.loan.model;

public class Loan{

private String number;

private String type;

private int loan;

private int emi;

private int tenure;

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

{

this.emi=emi;

this.loan=loan;

this.number=number;

this.tenure=tenure;

this.type=type;

}

public String getNumber()

{

return number;

}

public String getType()

{

return type;

}

public int getEMI()

{

return emi;

}

public int getTenure()

{

return tenure;

}

public int getLoan()

{

return loan;

}

}

**Loan Controller – LoanController.java**

**Code:**

package com.cognizant.loan.controller;

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

import com.cognizant.loan.model.\*;

*@RestController*

*@RequestMapping*("/loans")

public class LoanController{

*@GetMapping*("/{number}")

public Loan getLoan(*@PathVariable* String number)

{

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

}

}

**LoanApplication.java Running on port 5000**

**Code:**

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);

}

}

**Output:**

Account:

A screenshot of a computer

AI-generated content may be incorrect.

Loan:  
A screenshot of a computer

AI-generated content may be incorrect.

**Handson - Create Eureka Discovery Server and Register Microservices**

**1. Created eureka-discovery-server**

**Application code:**

package com.cognizant.eureka\_discovery\_server;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

*@SpringBootApplication*

*@EnableEurekaServer*

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.*run*(EurekaDiscoveryServerApplication.class, args);

}}

**2. Registering MicroServices which are created in previous Handson:**

**AccountApllication Code:**

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

*@SpringBootApplication*

*@EnableDiscoveryClient*

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.*run*(AccountApplication.class, args);

}

}

**LoanApplication Code:**

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

*@SpringBootApplication*

*@EnableDiscoveryClient*

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.*run*(LoanApplication.class, args);

}

}

**Added the below line in Application Properties of both microservices to register them under eureka-discovery-server**

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

**Output after accessing** [**http://localhost:8761/eureka**](http://localhost:8761/eureka)

