Creating Microservices for account and loan

Part 1: Create Account Microservice

AccountController.java

package com.cognizant.account;

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

*@RestController*

public class AccountController {

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

public Account getAccountDetails(*@PathVariable* String number) {

return new Account("00987987973432", "savings", 234343);

}

}

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; }

}

AccountApplication.java

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AccountApplication {

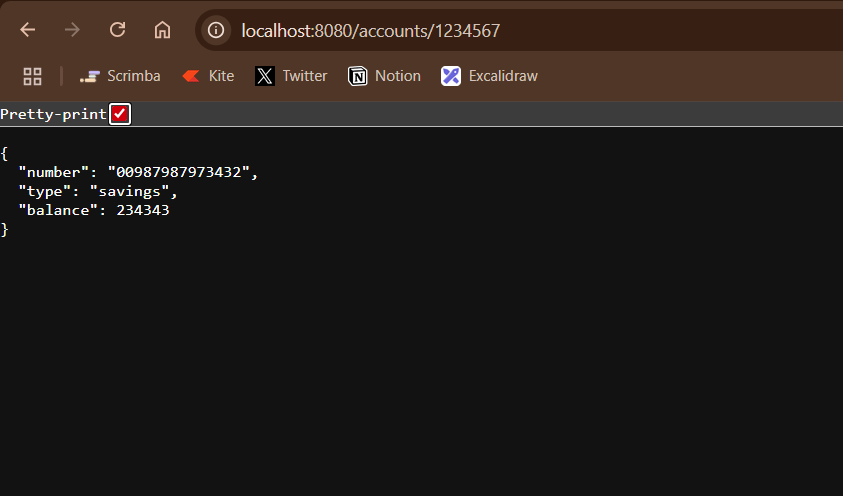
public static void main(String[] args) {

SpringApplication.run(AccountApplication.class, args);

}

}

Output :



Part 2: Create Loan Microservice

application.properties :

spring.application.name=loan

server.port=8081

LoanController.java :

package com.cognizant.loan;

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

*@RestController*

public class LoanController {

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

public Loan getLoanDetails(*@PathVariable* String number) {

return new Loan("H00987987972342", "car", 400000, 3258, 18);

}

}

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.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 int getLoan() { return loan; }

public int getEmi() { return emi; }

public int getTenure() { return tenure; }

}

Output :

