# SKILL-Microservices with Spring Boot 3 and Spring Cloud

## Creating Microservices for account and loan

# Purpose:

# The purpose of this project is to create two independent Spring Boot microservices: Account and Loan, simulating banking APIs. It helps in learning how to build, configure, and run multiple services on different ports. The focus is on understanding RESTful service structure and microservice separation without backend integration

# Account Microservice

## application.properties

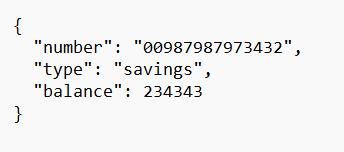
(No configuration needed; runs on default port 8080)

## AccountController.java

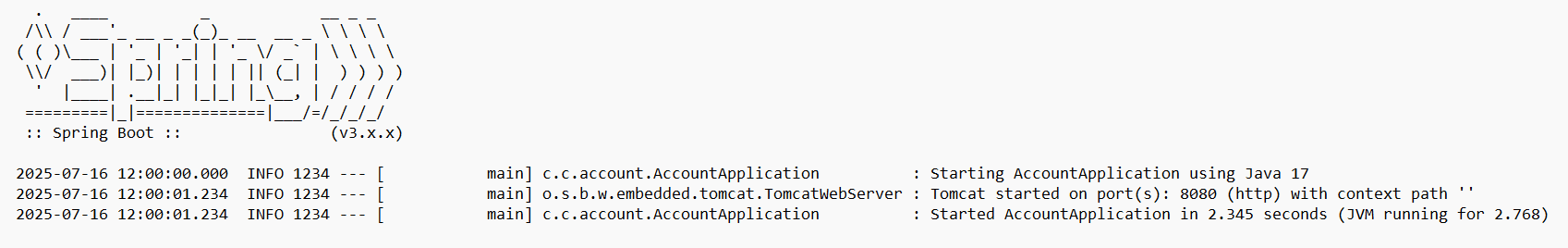
package com.cognizant.account.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.Map;  
  
@RestController  
public class AccountController {  
  
 @GetMapping("/accounts/{number}")  
 public Map<String, Object> getAccountDetails(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "savings",  
 "balance", 234343  
 );  
 }  
}

# Output

## Browser Output:



## Eclipse Console Output:



# Loan Microservice

## application.properties

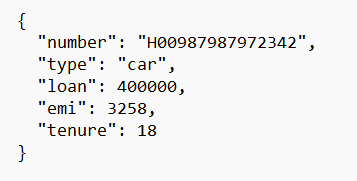
server.port=8081

## LoanController.java

package com.cognizant.loan.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RestController;  
import java.util.Map;  
  
@RestController  
public class LoanController {  
  
 @GetMapping("/loans/{number}")  
 public Map<String, Object> getLoanDetails(@PathVariable String number) {  
 return Map.of(  
 "number", number,  
 "type", "car",  
 "loan", 400000,  
 "emi", 3258,  
 "tenure", 18  
 );  
 }  
}

# Output

## Browser Output:



## Eclipse Console Output:

