**Creating Microservices for Account and Loan**

**Objective:**

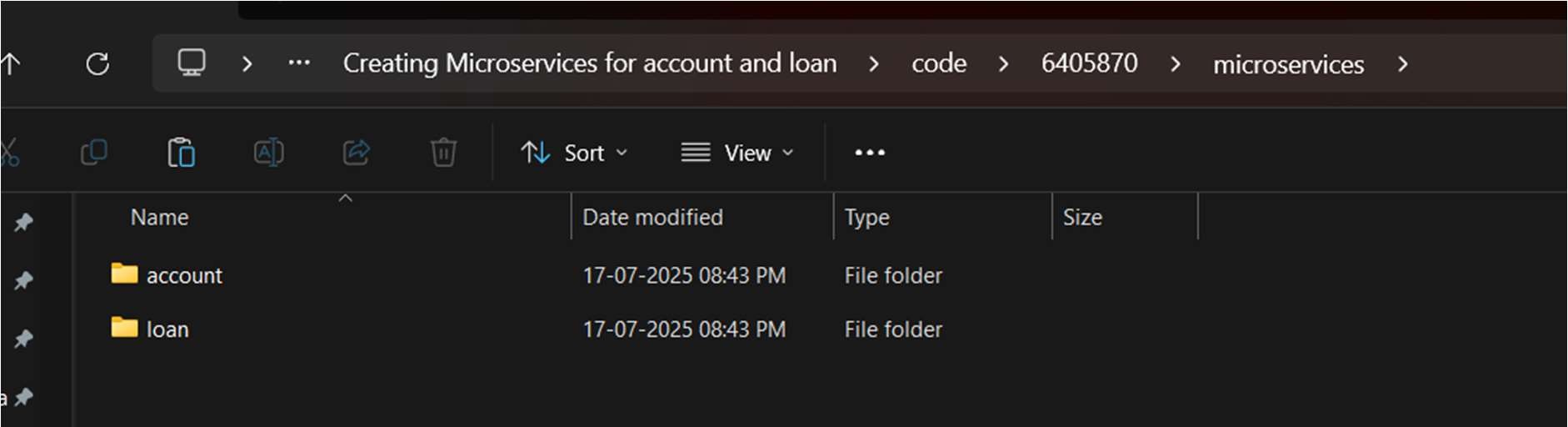
To create two separate microservices using **Spring Boot**:

* One for **Account**
* One for **Loan**

Each microservice will have its own Maven project and will return dummy data using a REST API. No database is used.

**Project Setup:**

* Created a folder structure:



* Used start.spring.io to generate two Spring Boot projects:
  + **account**
  + **loan**
* Selected dependencies:
  + Spring Web
  + Spring Boot DevTools
* Built each project using:
* mvn clean package

**1. Account Microservice**

**AccountController Code:**

package com.cognizant.account.controller;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Account getAccount(@PathVariable String number) {

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

}

static class Account {

public String number;

public String type;

public double balance;

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

this.number = number;

this.type = type;

this.balance = balance;

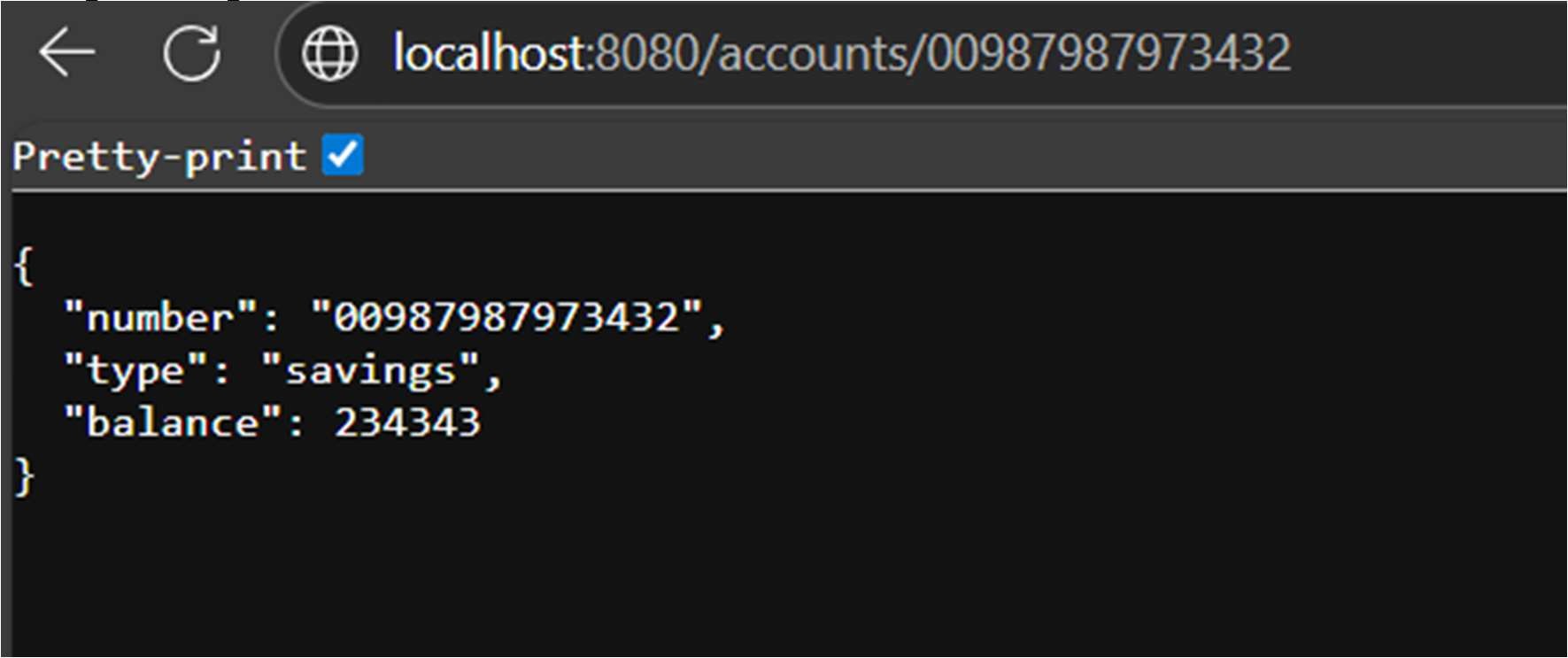
}

}

}

**Tested URL:**

<http://localhost:8080/accounts/00987987973432>

**Sample Output:**

**2. Loan Microservice**

**Port Change:**

In application.properties file:

server.port=8081

**LoanController Code:**

package com.cognizant.loan.controller;

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

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Loan getLoan(@PathVariable String number) {

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

}

static class Loan {

public String number;

public String type;

public double loan;

public double emi;

public 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;

}

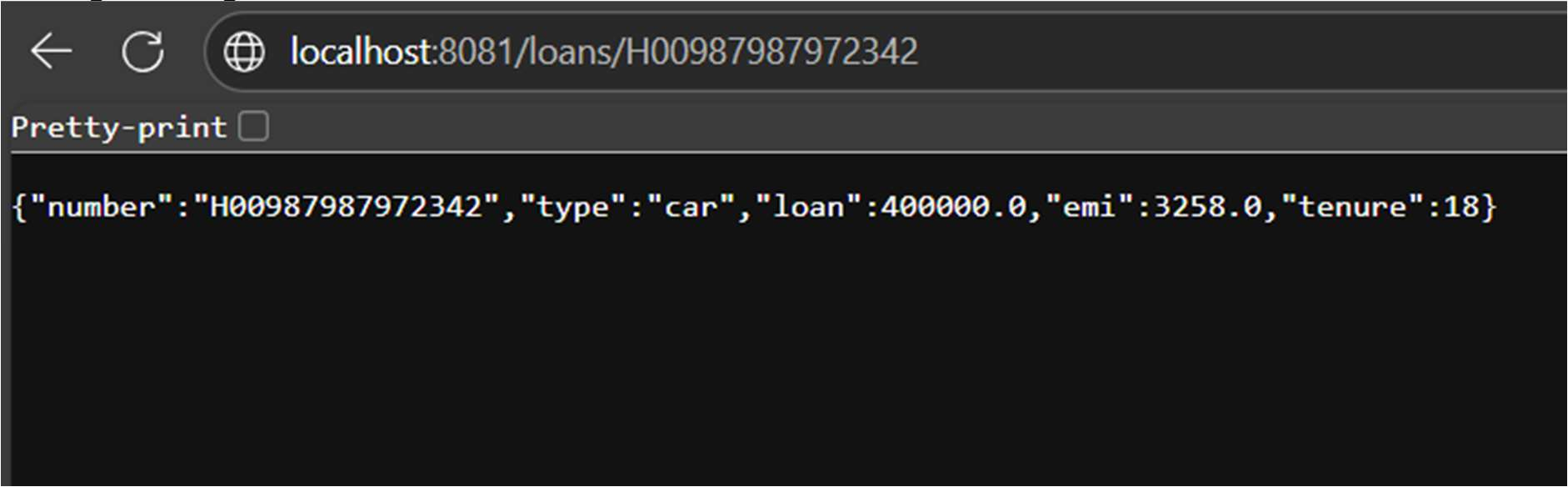
}

}

**Tested URL:**

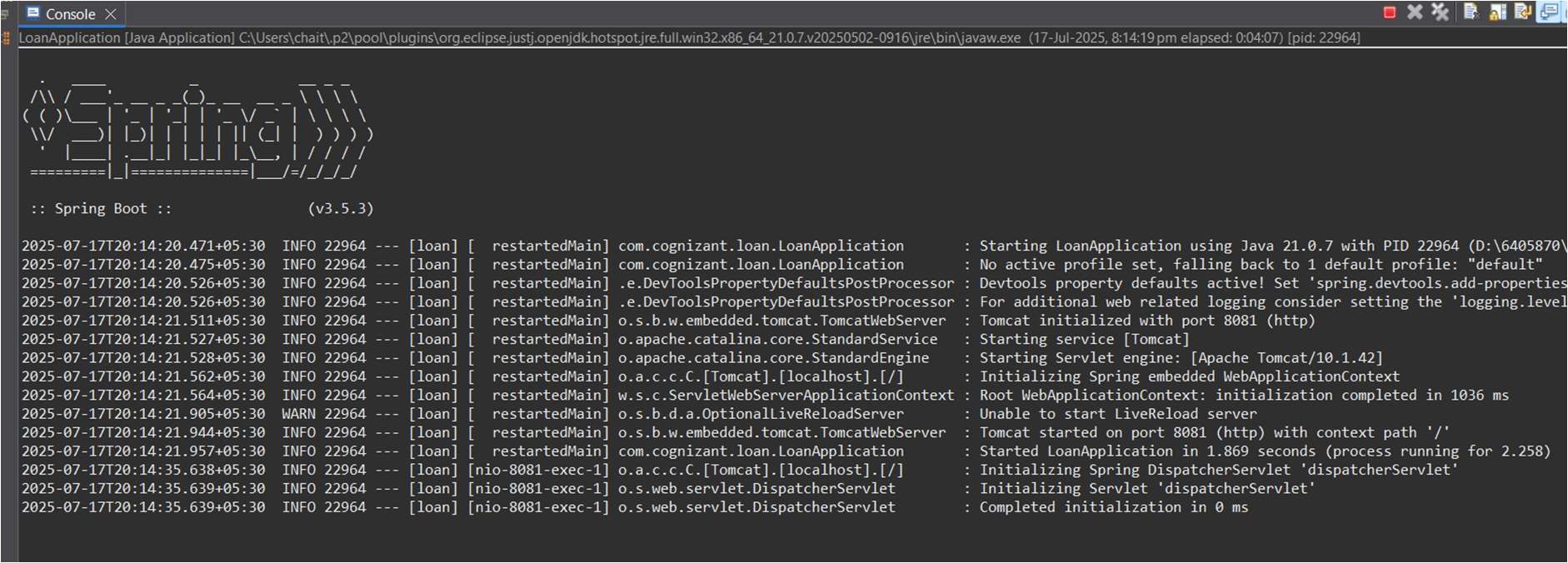
<http://localhost:8081/loans/H00987987972342>

**Sample Output:**



**Conclusion:**

* Created two working microservices.
* Used GET APIs to return sample data.
* Handled port conflict using application.properties.
* Successfully tested both services on different ports.

**Console Output Screenshots:**

