**WEEK 5** **MICROSERVICES**

**Creating**

**Microservices for Account and Loan**

**Objective**

To design and implement two independent Spring Boot-based microservices:

* One for handling **Account** information
* Another for handling **Loan** information

**Tools & Technologies Used**

* Java
* Spring Boot
* Spring Web
* Maven
* Eclipse IDE
* Postman / Browser
* Spring Initializr

**Folder Structure**

📁D:\EmployeeID-01\microservices  
    📁 account  
    📁 loan

**Microservice 1 – Account Service**

**Spring Initializr Configuration**

* Group: com.cognizant
* Artifact: account
* Dependencies: Spring Boot DevTools, Spring Web

**Endpoint Specification**

* **Method**: GET
* **URL**: http://localhost:8090/accounts/{number}
* **Sample Request**:  
  http://localhost:8090/accounts/00987987973432
* **Sample Response**:

{

"number": "00987987973432",

"type": "savings",

"balance": 234343

}

**Code – AccountController.java**

package com.cognizant.account.controller;

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

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

);

}

}

**Port Configuration**

src/main/resources/application.properties:

spring.application.name=account

server.port=8090

**Microservice 2 – Loan Service**

**Spring Initializr Configuration**

* Group: com.cognizant
* Artifact: loan
* Dependencies: Spring Boot DevTools, Spring Web

**Endpoint Specification**

* **Method**: GET
* **URL**: http://localhost:8081/loans/{number}
* **Sample Request**:  
  <http://localhost:8081/loans/H00987987972342>
* **Sample Response**:

{

"number": "H00987987972342",

"type": "car",

"loan": 400000,

"emi": 3258,

"tenure": 18

}

**Code – LoanController.java**

package com.cognizant.loan.controller;

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

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

);

}

}

**Port Configuration**

src/main/resources/application.properties:

spring.application.name=loan

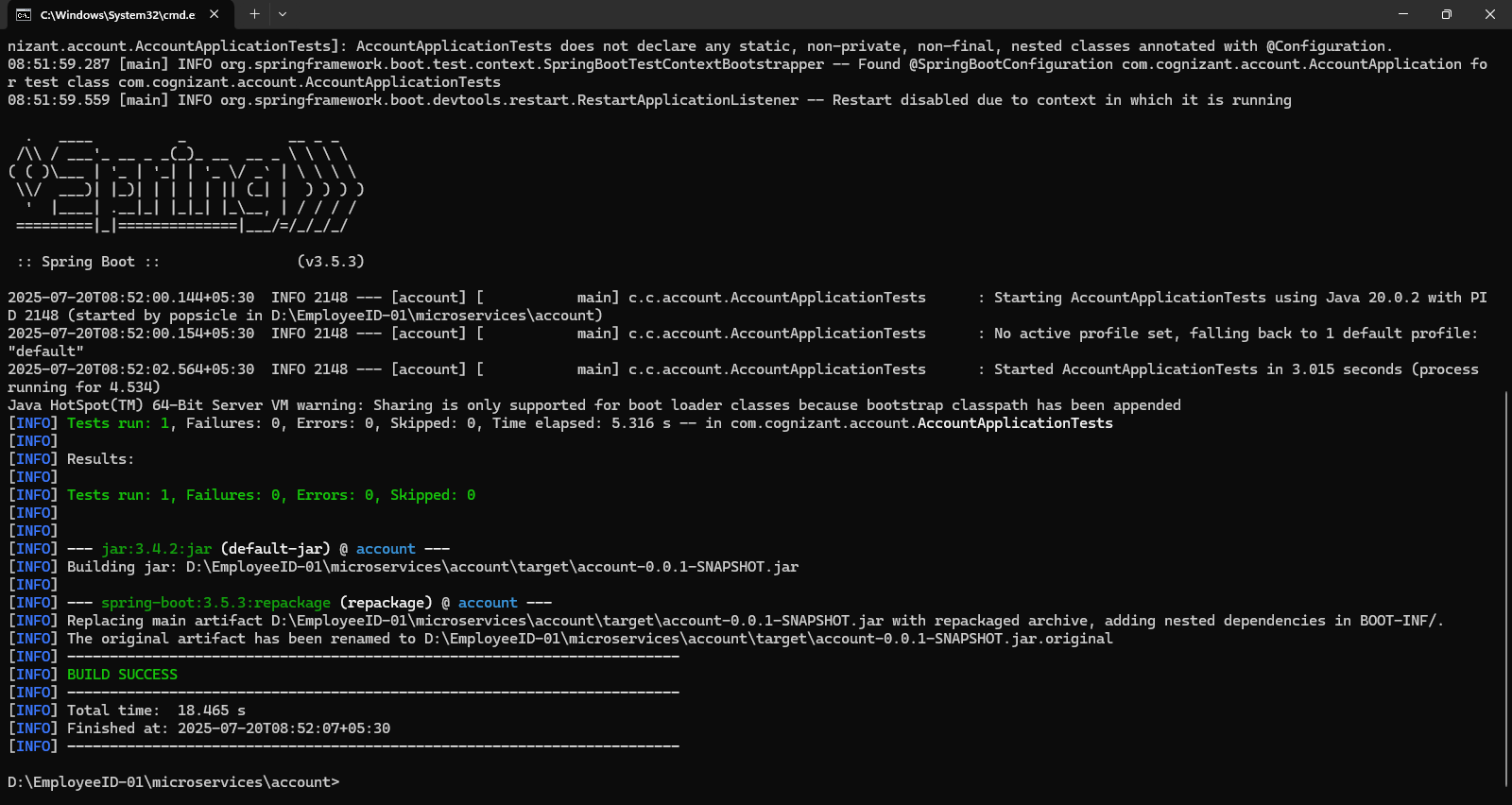
server.port=8081

**Testing the Services**

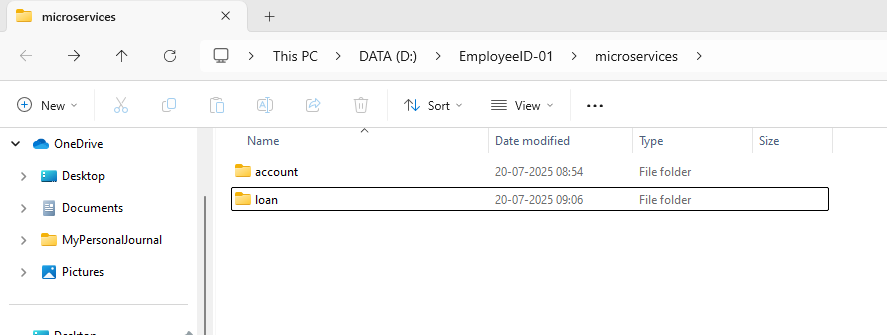
* **Account Service** was tested at:  
  http://localhost:8090/accounts/00987987973432
* **Loan Service** was tested at:  
  http://localhost:8081/loans/H00987987972342

**SCREENSHOTS**

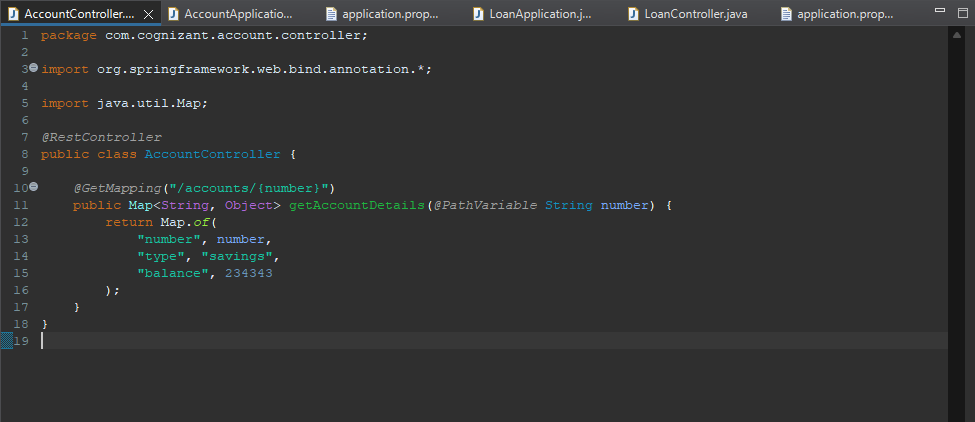
**SPRING BUILD SUCCESS**



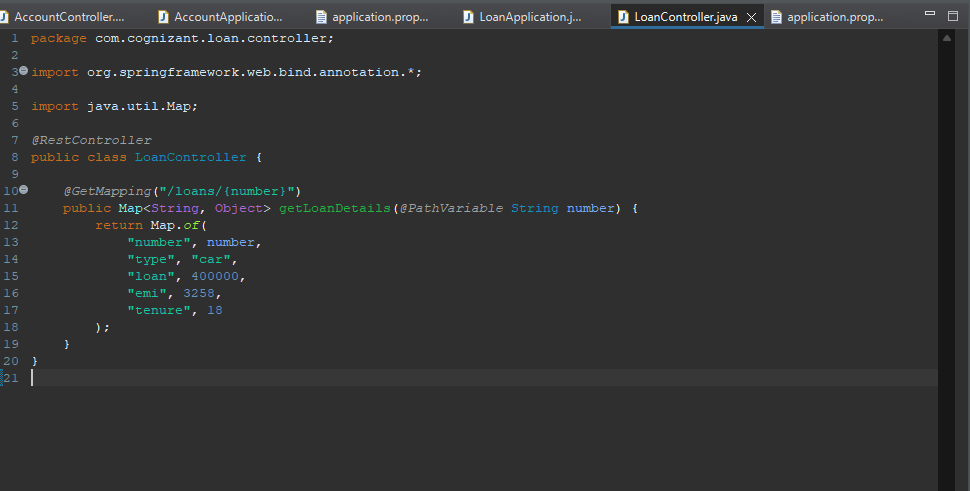
**Folder structure in D drive:**

****

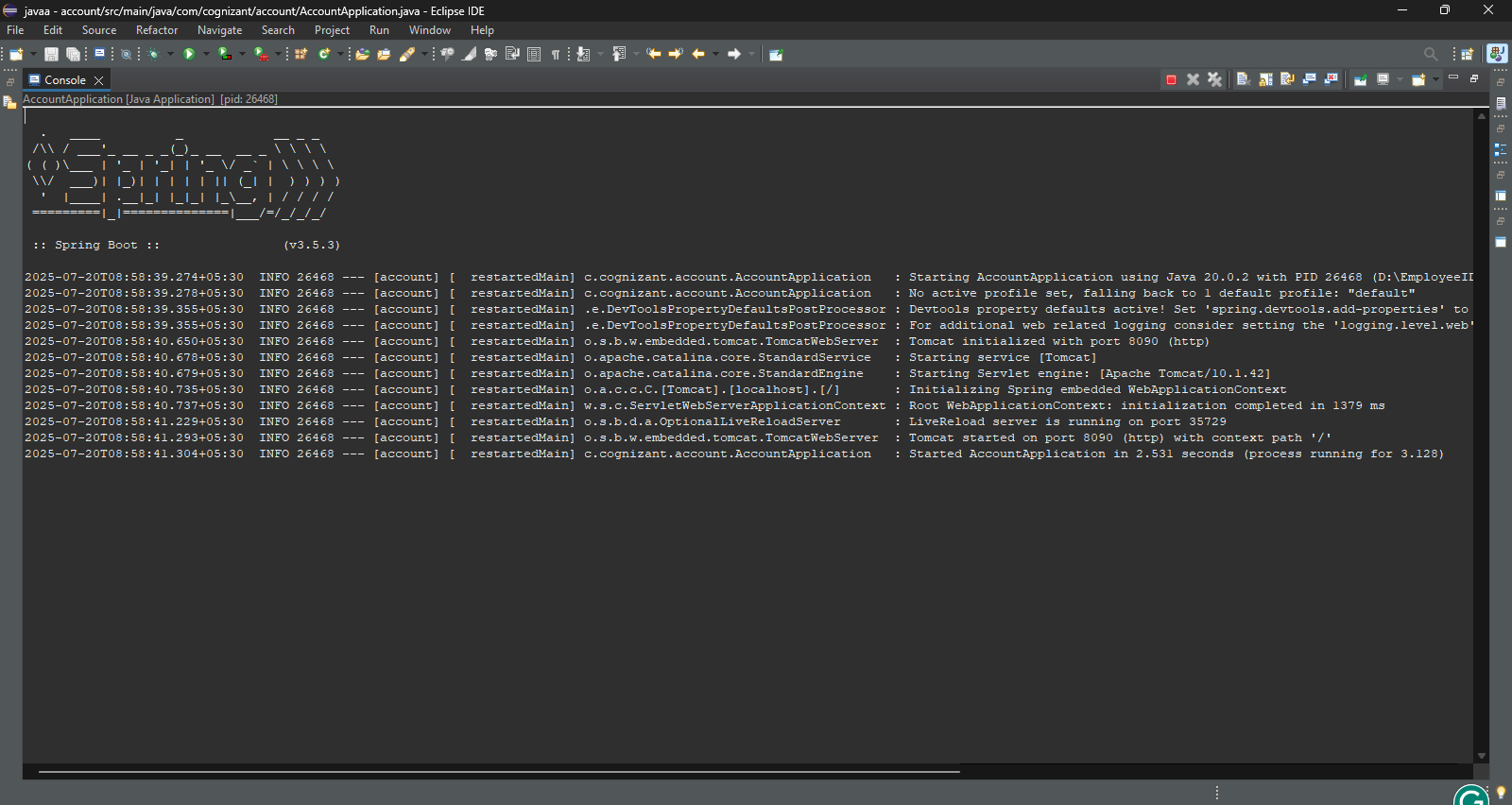
**AccountController.java**

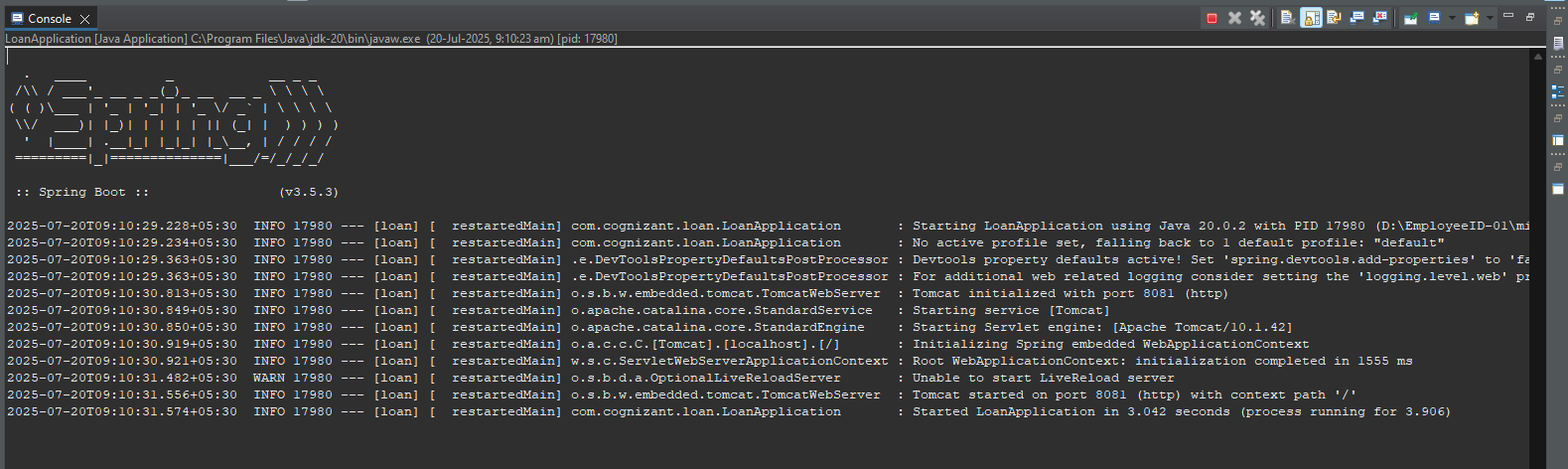
****

**LoanController.java**

****

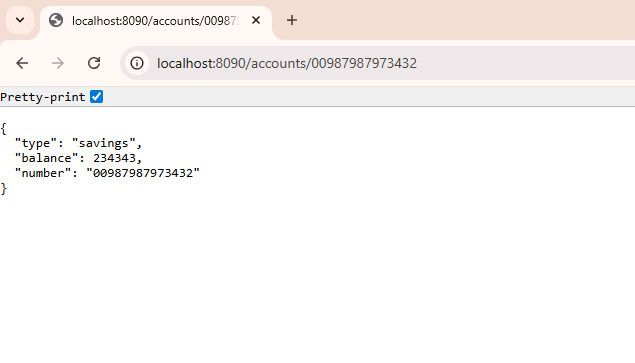
**Console output:**

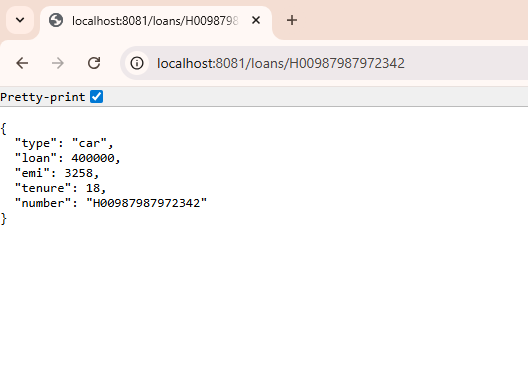
****

****

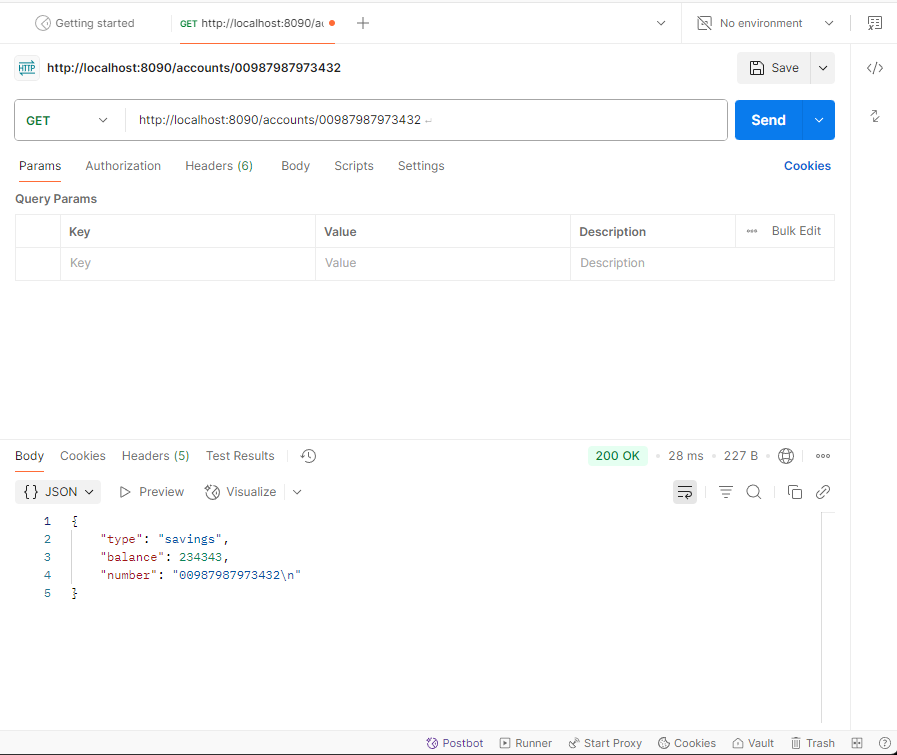
**Chrome server host:**

**Accounts:**

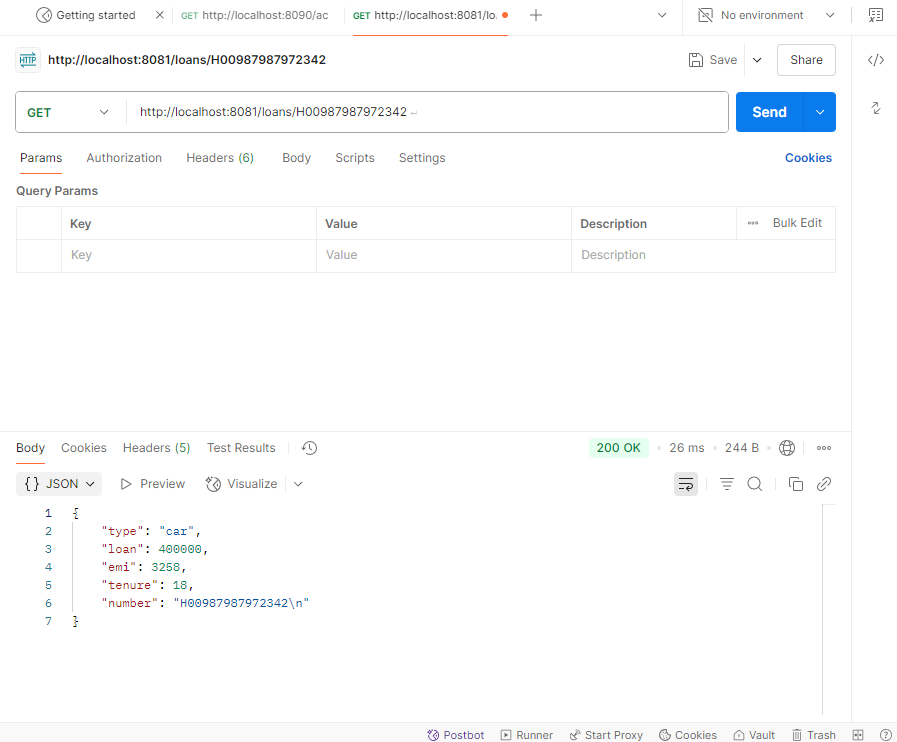
**Loan:**

****

**Postman output Accounts:**

****

**Postman output Loans:**

****

**Conclusion**

Two independent microservices were created and run on different ports:

* account on port 8090
* loan on port 8081

Each service exposed a REST API and returned mock data without any database. This exercise helped demonstrate the microservices architecture and the use of Spring Boot to develop scalable RESTful applications.