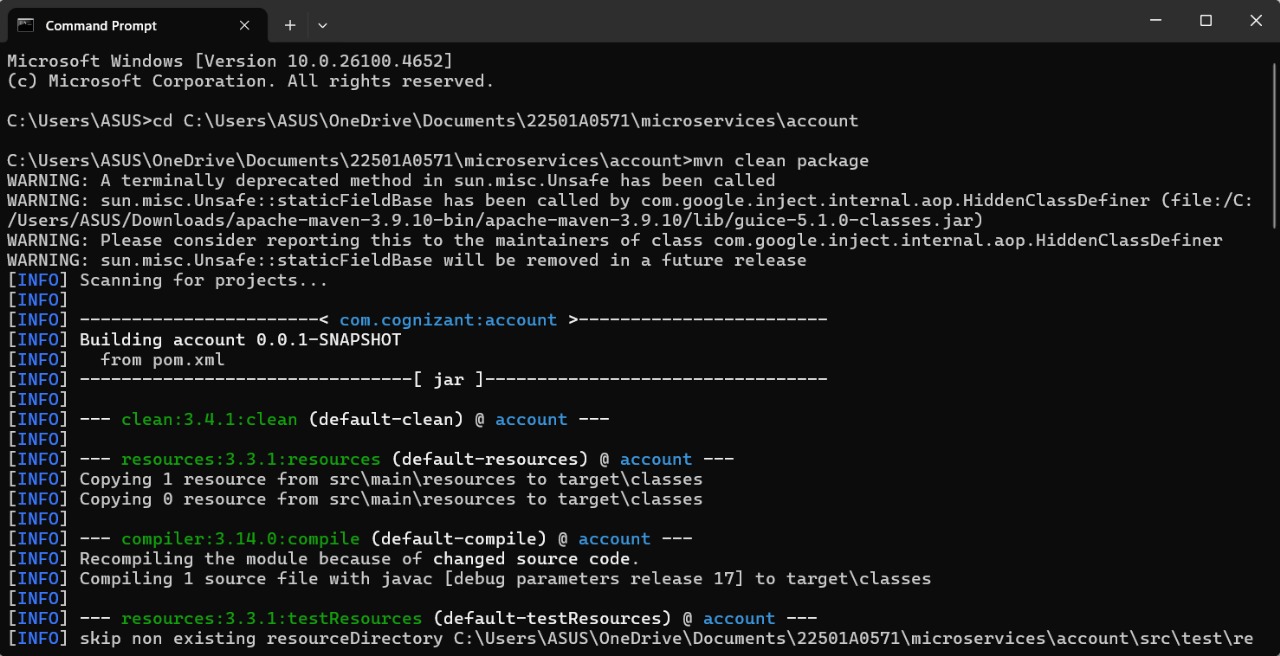
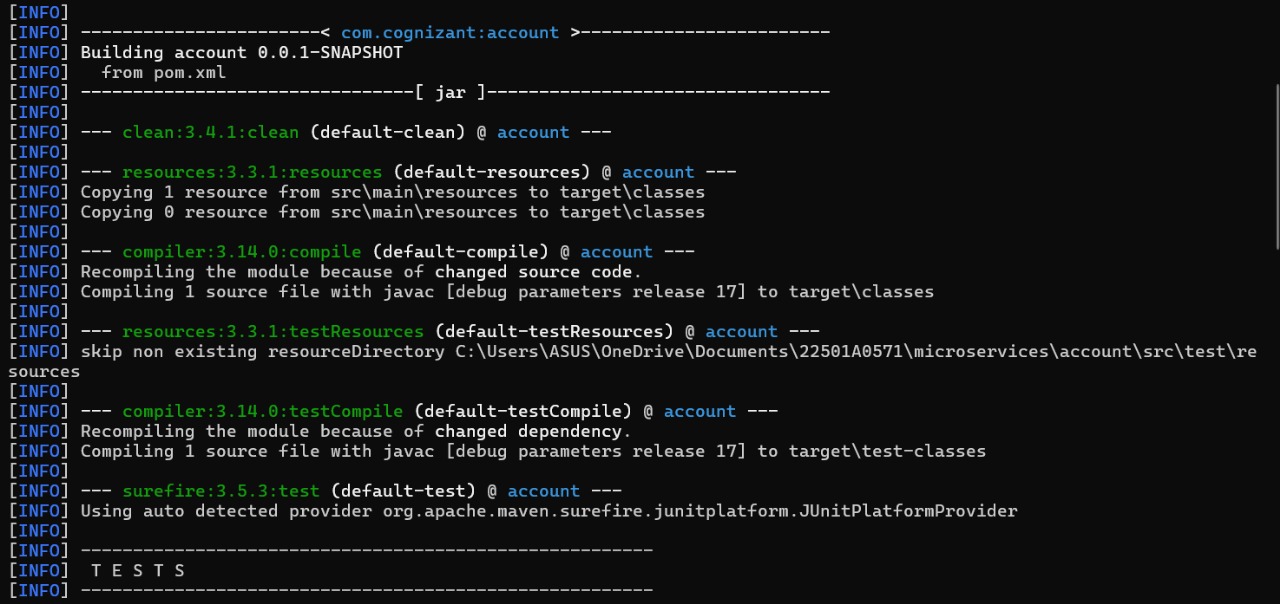
**MICROSERVICES WITH SPRING BOOT 3 AND SPRING CLOUD**

**Creating Microservices for account and loan**

**Account Microservice**

* Create folder with employee id in D: drive
* Create folder named 'microservices' in the new folder created in previous step. This folder will contain all the sample projects that we will
* create for learning microservices.
* Open https://start.spring.io/ in browser
* Enter form field values as specified below:
  + Group: com.cognizant
  + Artifact: account
* Select the following modules
* Developer Tools > Spring Boot DevTools
* Web > Spring Web
* Click generate and download the zip file
* Extract 'account' folder from the zip and place this folder in the 'microservices' folder created earlier
* Open command prompt in account folder and build using **mvn clean package** command





**After importing the project in Eclipse:**

**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);

}

}

**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> getAccount(@PathVariable String number) {

return Map.of(

"number", number,

"type", "savings",

"balance", 480000

);

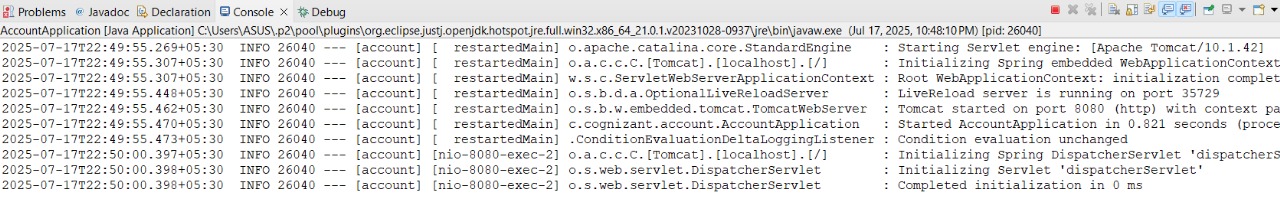
}

}

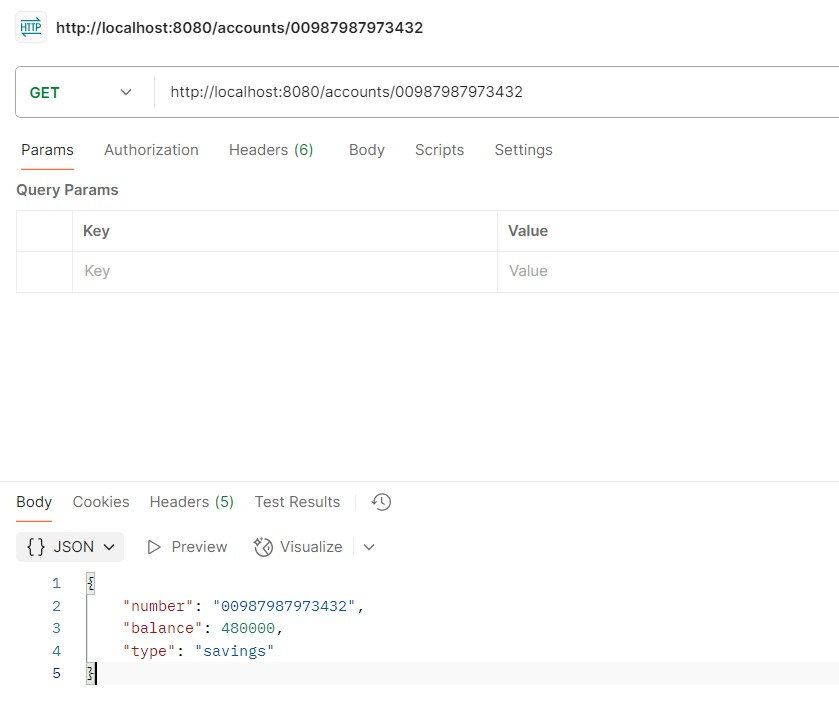
Run the application:

* Go to the AccountApplication.java class
* Right-click > **Run As > Java Application**

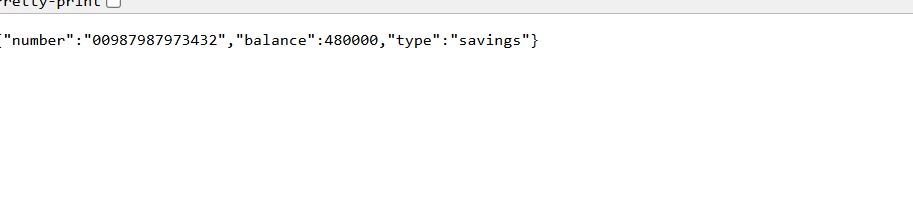
**OUTPUT:**



**In postman:**



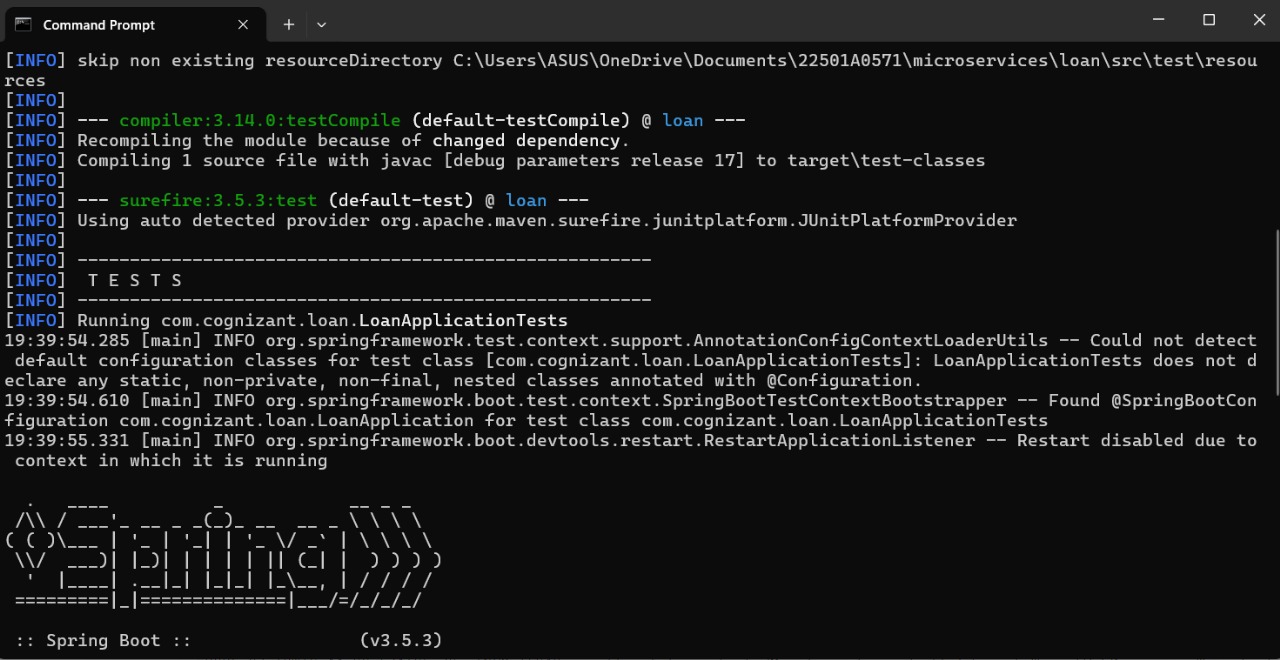
**In Browser:**

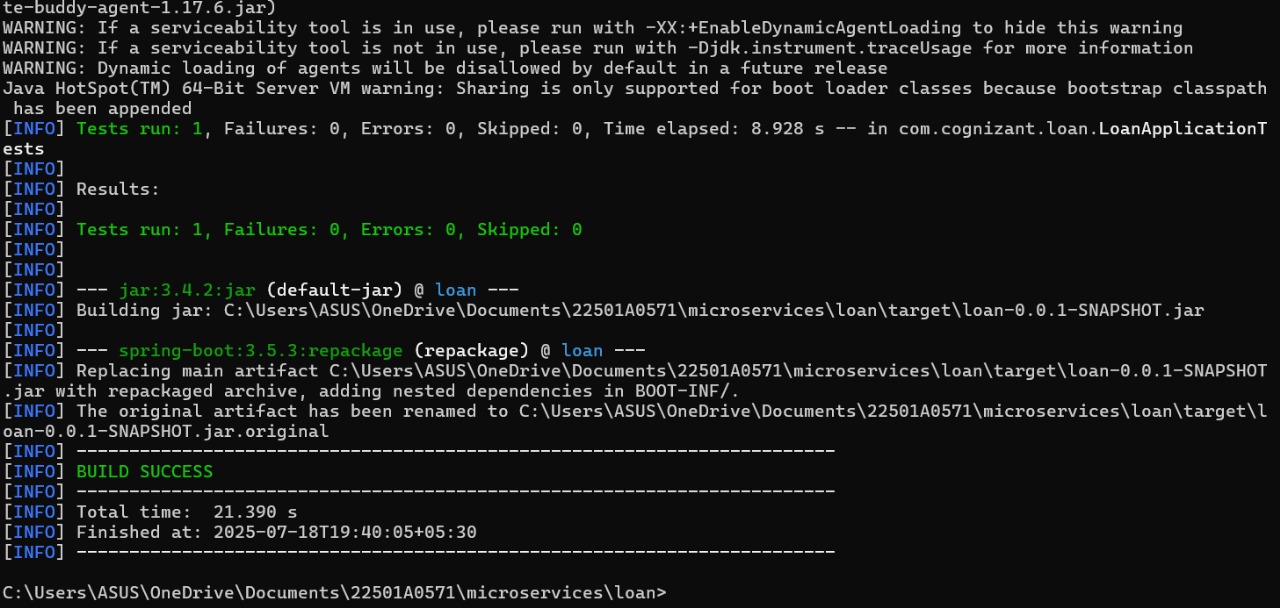


**Loan Microservice**

**Generate Spring Boot Project for Loan**

* Go to <https://start.spring.io/>
* Fill in the form:
  + Group: com.cognizant
  + Artifact: loan
* Select Dependencies:
  + Spring Boot DevTools
  + Spring Web
* Click Generate, download the ZIP file.
* Extract 'loan' folder from the zip and place this folder in the 'microservices' folder created earlier
* Open command prompt in account folder and build using **mvn clean package** command





**After importing project n Eclipse:**

**LoanApplication.java**

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);

}

}

**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> getLoan(@PathVariable String number) {

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

"tenure", 18

);

}

}

* Launching this application by having account service already running
* This launch will fail with error that the bind address is already in use
* The reason is that each one of the service is launched with default port number as 8080.
* Account service is already using this port and it is not available for loan service.
* Include "server.port" property with value 8081 and try launching the application

**server.port=8081**

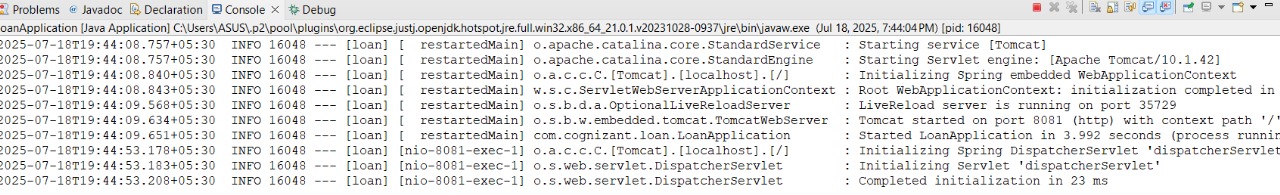
* Test the service with 8081 port

Now we have two microservices running on different ports.

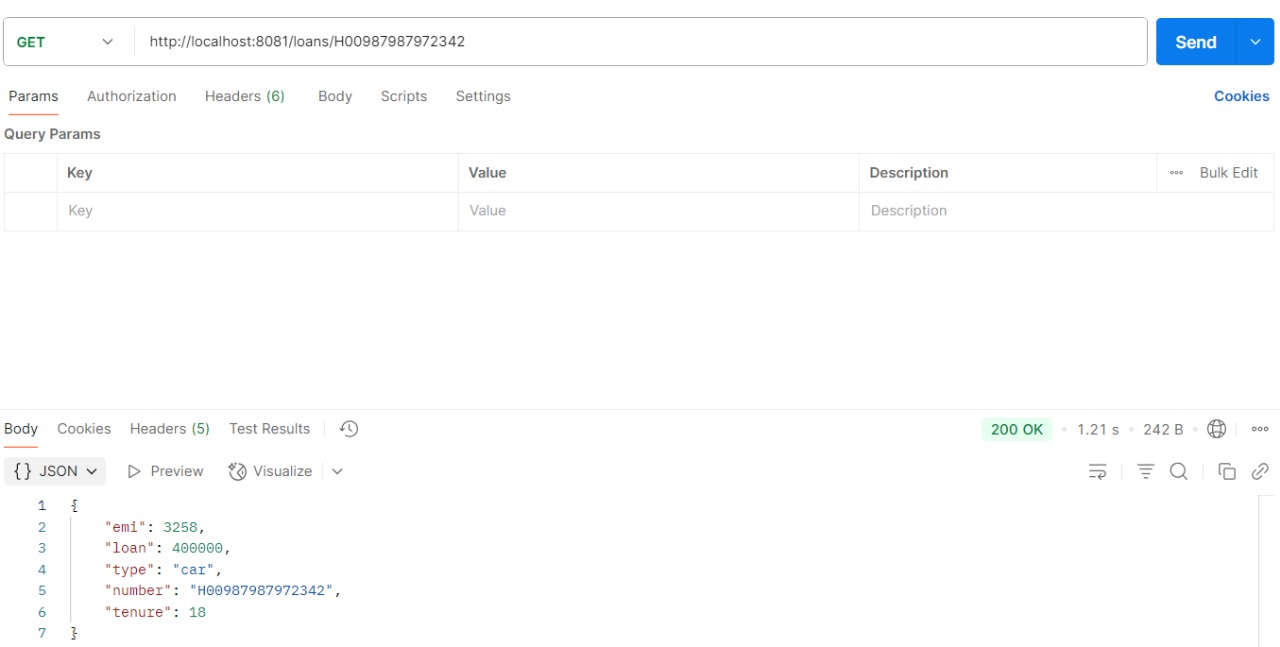
Run the application:

* Go to the LoanApplication.java class.
* Right-click > **Run As > Java Application**

**OUTPUT:**



**In postman:**



**In browser:**

