**WEEK – 05**

**Microservices**

**Microservices with API gateway**

**1.Creating Microservices for account and loan**

**Solution:**

**AccountController.java:**

package com.cognizant.account;

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

);

}

}

**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);

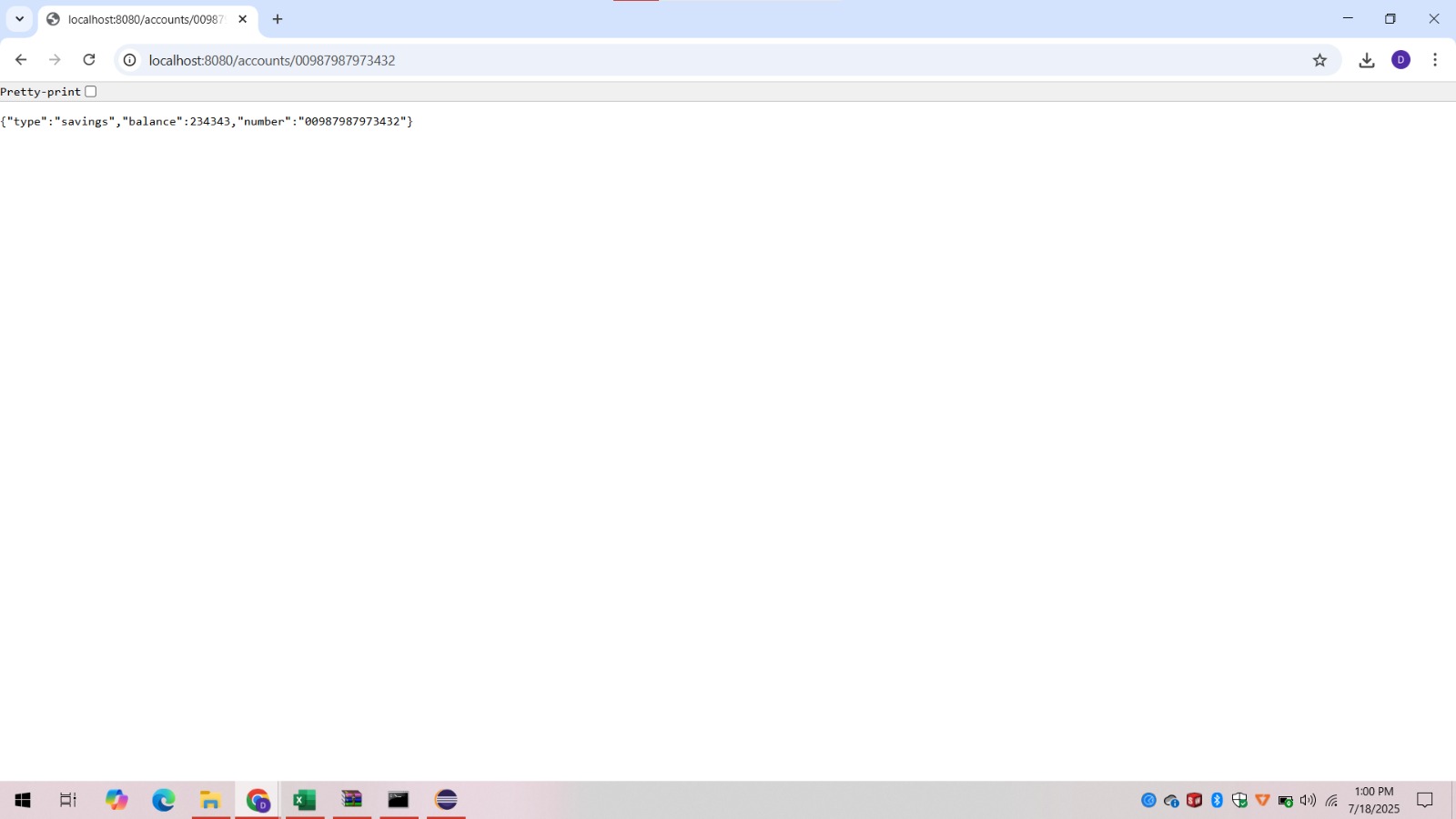
}

}

**Output:**

**(For Account)**





**Explanation: (For Account)**

* The Spring Boot project for the Account microservice was downloaded as a ZIP file from [start.spring.io](https://start.spring.io).
* After extraction, the folder was initially placed as D:\123456\microservices\account\account, creating an unnecessary nested directory.
* This structure caused navigation issues in Command Prompt and confusion during project import in Eclipse.
* To fix this, the inner account folder content (like pom.xml, src/, etc.) was moved up to D:\123456\microservices\account.
* The corrected structure ensures Maven can build the project properly and Eclipse can recognize it as a valid Maven project.
* A clean folder structure is essential for effective project setup, build, and IDE integration.

**LoanController.java:**

package com.cognizant.loan;

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", "home loan",

"amount", 500000

);

}

}

**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);

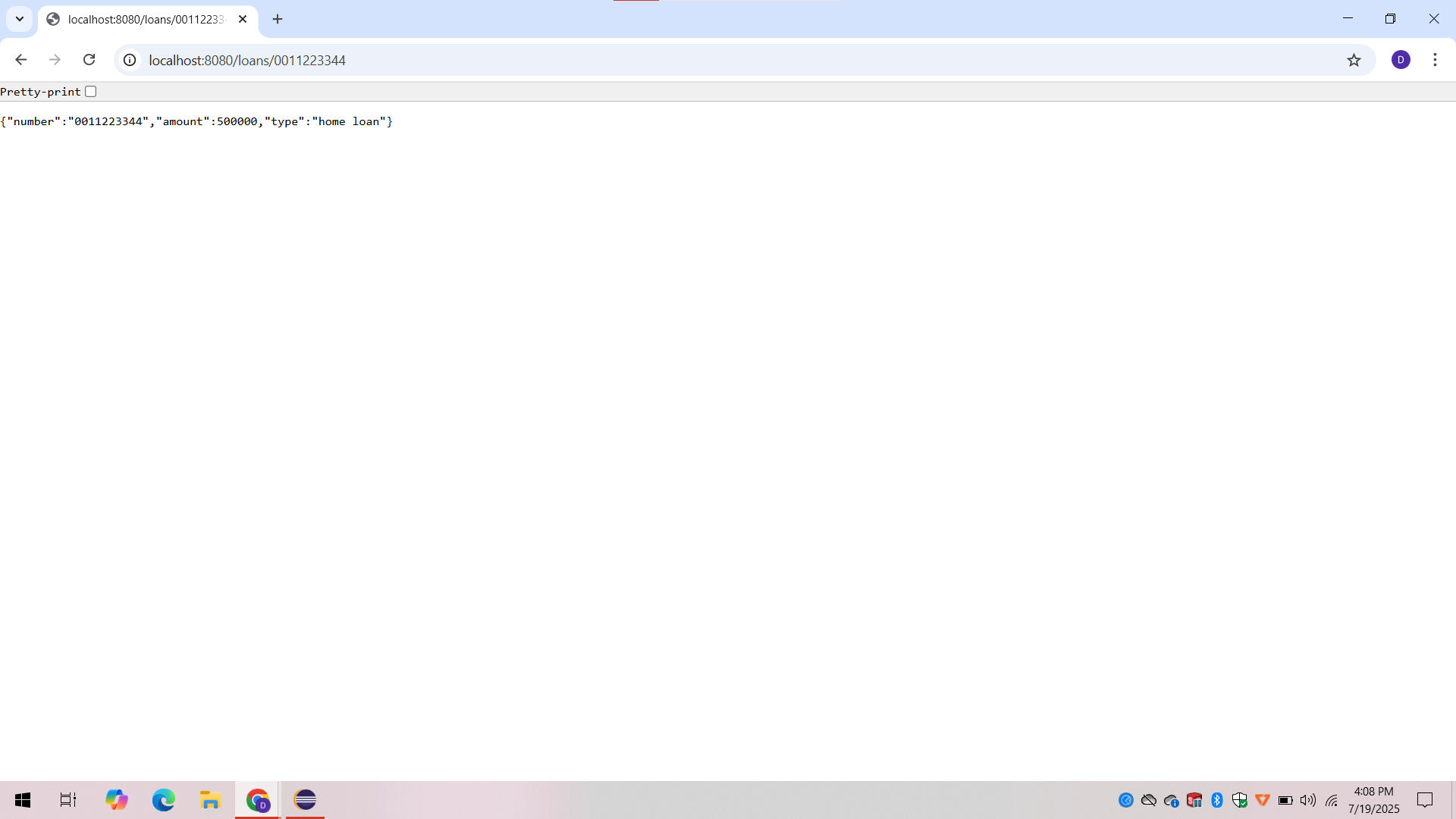
}

}

**Output:**

**(For Loan Services)**





**Explanation:(For LoanServices)**

* A separate Spring Boot project for the Loan microservice was generated using [start.spring.io](https://start.spring.io) with group as com.cognizant and artifact as loan.
* LoanApplication.java is the **main class** that bootstraps the Loan Microservice.
* It is located in the package com.cognizant.loan and created automatically by Spring Initializr.
* The class is annotated with @SpringBootApplication, which enables component scanning, auto-configuration, and Spring configuration.
* The main() method uses SpringApplication.run() to start the embedded Tomcat server and initialize the application.
* This class acts as the **entry point** of the microservice and runs as a standalone Java application.