**Digital Nurture 4.0 – Week 5**

**Mandatory hands-on**

**Creating Microservices for account and loan:**

In this hands on exercises, we will create two microservices for a bank. One microservice for handing accounts and one for handling loans.Each microservice will be a specific independent Spring RESTful Webservice maven project having it's own pom.xml. The only difference is that, instead of having both account and loan as a single application, it is split into two different applications. These webservices will be a simple service without any backend connectivity.

**Objective:**

To Create a Microservices for an account and loan Using the Webservices.

**Steps:**

**i)Creating Account**

**Step1:Creating a Project File**

Go to your D: drive.

Create a folder with your Employee ID.  
*Example*: D:\1234567

Inside this folder, create a subfolder named microservices.

**Step 2: Create Account Microservice**

Open: <https://start.spring.io>

Fill the form:

* Group: com.cognizant
* Artifact: account

Choose dependencies:

* Spring Boot DevTools (under *Developer Tools*)
* Spring Web (under *Web*)

Click Generate and download the account.zip file.

**Import into Eclipse**

1. Open Eclipse → File → Import → Existing Maven Projects
2. Select the account folder from D:\1234567\microservices
3. Click Finish

**Create Controller:**

**src/main/java/com/cognizant/account, create a controller:**

package com.cognizant.account;

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

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Account getAccount(@PathVariable String number) {

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

}

}

class Account {

private String number;

private String type;

private double balance;

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

this.number = number;

this.type = type;

this.balance = balance;

}

public String getNumber() { return number; }

public String getType() { return type; }

public double getBalance() { return balance; }

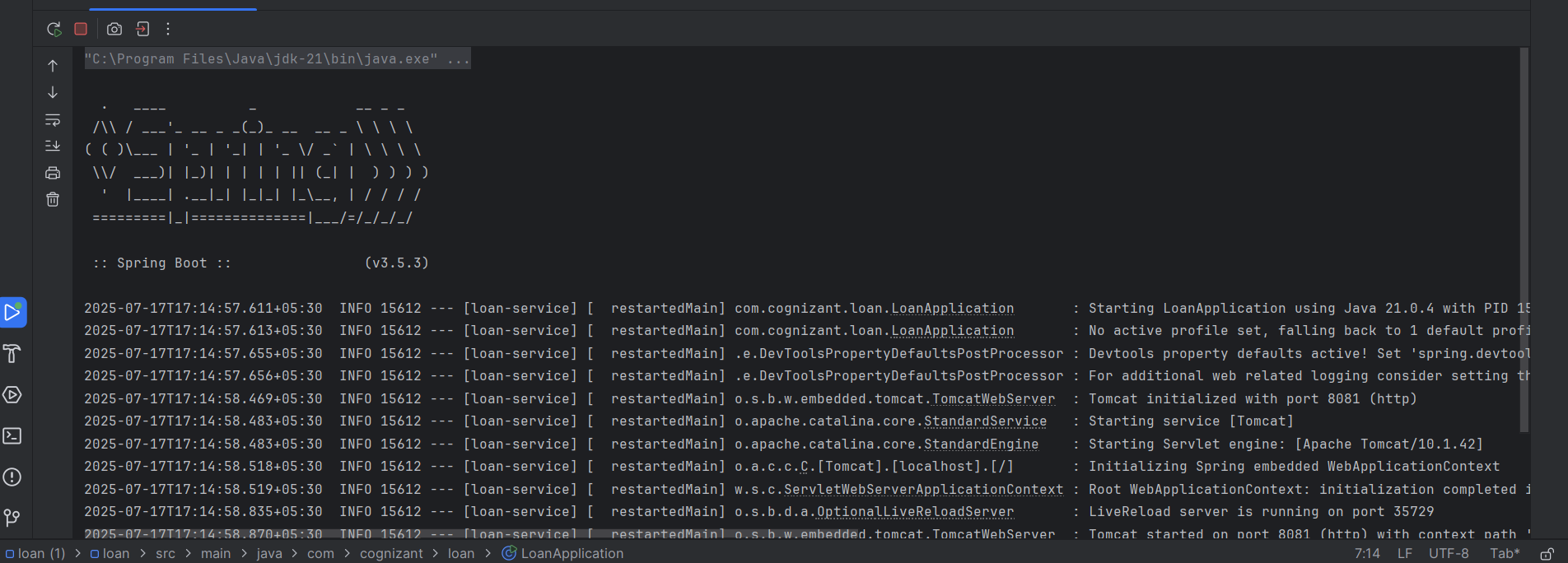
}

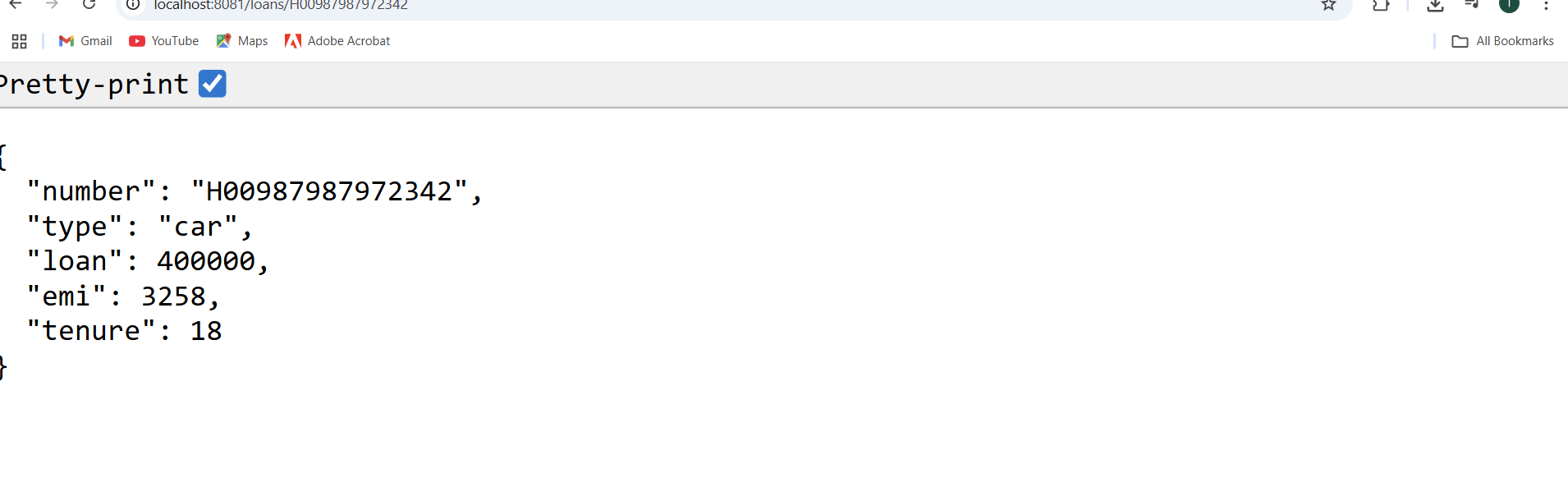
**Run the Application**

1. Find AccountApplication.java under com.cognizant.account
2. Right-click → Run
3. Open browser:

<http://localhost:8081/accounts/00987987973432>

**Output:**

****

****

**ii)Loan**

**Step 1: Create Loan Microservice**

Spring Initializr Again

1. File → New → Project → Spring Initializre
2. Group: com.cognizant
3. Artifact: loan
4. Dependencies:
   * Spring Web
   * Spring Boot DevTools
5. Click Finish

**Step 2:Create Controller**

**src/main/java/com/cognizant/loan, create LoanController**

package com.cognizant.loan;

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

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Loan getLoan(@PathVariable String number) {

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

}

}

class Loan {

private String number;

private String type;

private double loan;

private int emi;

private int tenure;

public Loan(String number, String type, double loan, int emi, int tenure) {

this.number = number;

this.type = type;

this.loan = loan;

this.emi = emi;

this.tenure = tenure;

}

public String getNumber() { return number; }

public String getType() { return type; }

public double getLoan() { return loan; }

public int getEmi() { return emi; }

public int getTenure() { return tenure; }

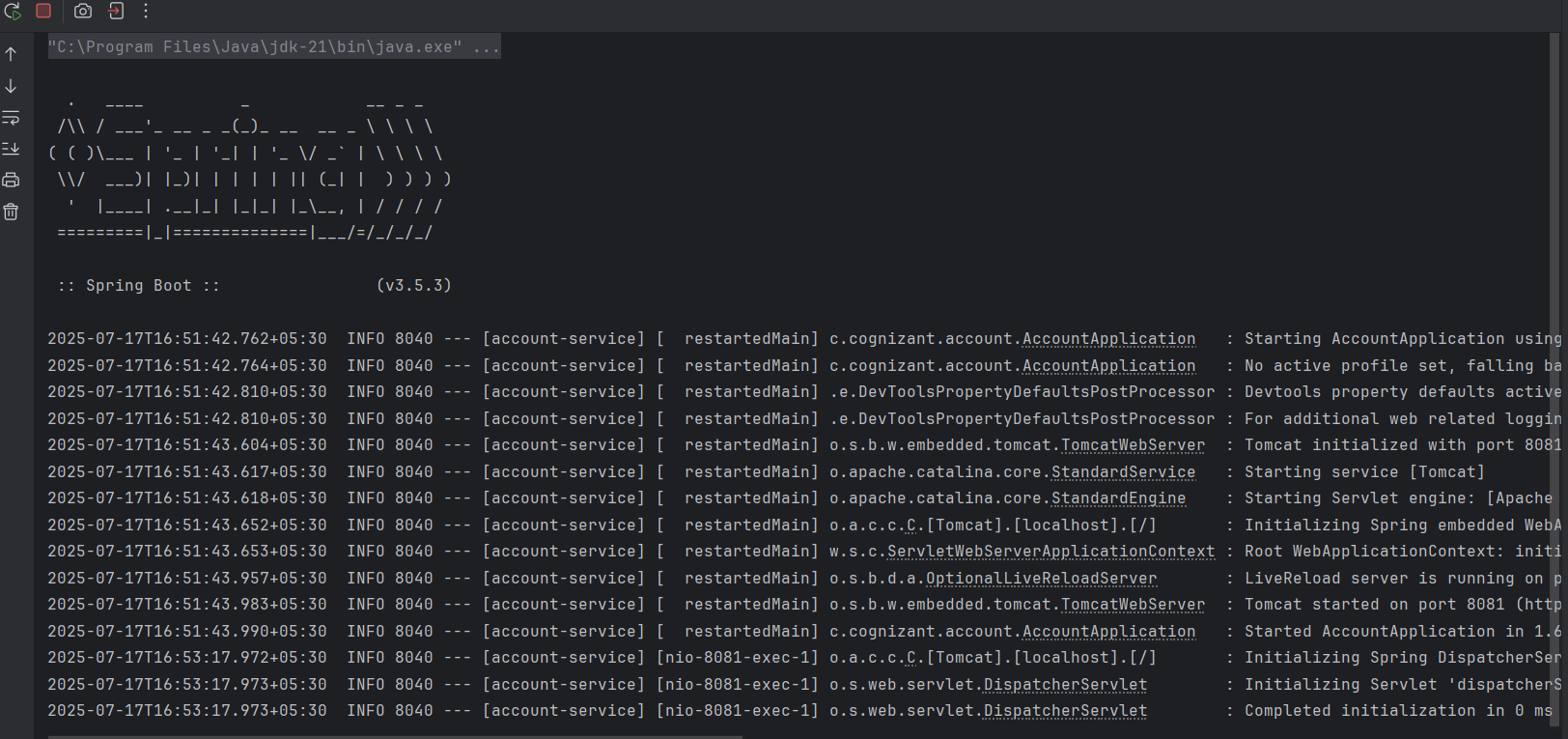
}

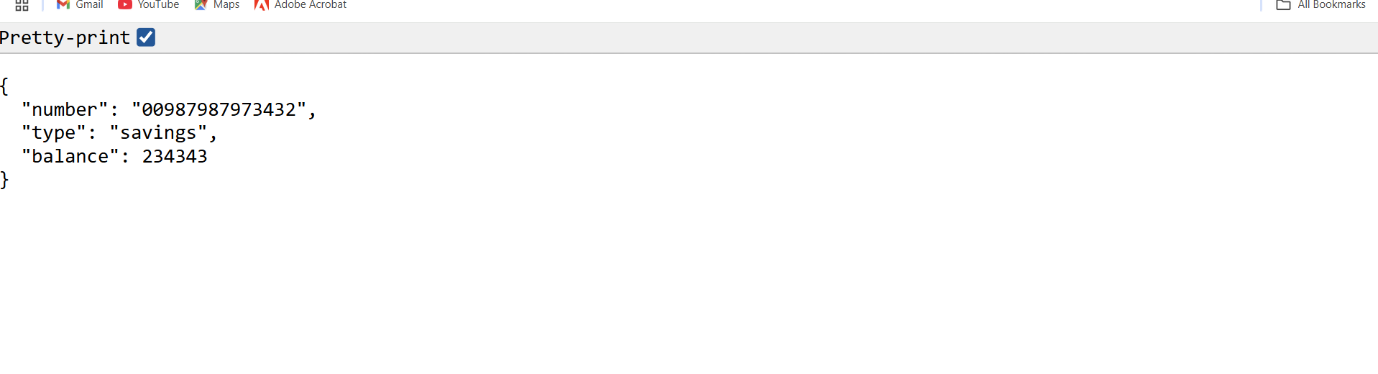
**Run Loan Application**

1. Make sure **Account** app is already running.
2. Run LoanApplication.java

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

**Output:**

****

****

**Additional important hands-on**

**Microservices with API gateway**

**2.Create Eureka Discovery Server and register**

**Step 1:**

Create a spring boot project named account.Refer below details

* + **Project:** Maven
  + **Language:** Java
  + **Spring Boot:** 3.2.4
  + **Group:** com.cognizant
  + **Artifact:** eurekaserver
  + **Name:** eurekaserver
  + **Package Name:** com.cognizant.eurekaserver
  + **Packaging:** Jar
  + **Java version:** 17

**Dependencies:**

* + Spring Boot DevTools
  + Eureka Server

**Step 2:**

Make changes on **pom.xml,application.properties,EurekaServerApplication.java.**

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>eureka-server</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>eureka-server</name>

<description>Eureka Server for service registry</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**application.properties**:

server.port=8761

spring.application.name=EUREKA-SERVER

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

**EurekaServerApplication.java**:

package com.cognizant.eurekaserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@SpringBootApplication

@EnableEurekaServer

public class EurekaServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaServerApplication.class, args);

}

}

**Step 3:**

Now run **EurekaServerApplication.java** as a java application and run <http://localhost:8761/> on your brower.After run account and loan server the final output will be shows in the end.

**Step 4:**

Create a spring boot project named account.Refer below details

* + **Project:** Maven
  + **Language:** Java
  + **Spring Boot:** 3.2.4
  + **Group:** com.cognizant
  + **Artifact:** account
  + **Name:** account
  + **Package Name:** com.cognizant.account
  + **Packaging:** Jar
  + **Java version:** 17

**Dependencies:**

* + Spring Boot DevTools
  + Spring Web
  + Eureka Discovery Client

**Step 5:**

Create the package named controller and also create class named **AccountController.java**.Alter **application.properties,AccountApplication.java** and ensure **pom.xml.**

**AccountController.java:**

package com.cognizant.account.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

@RequestMapping("/account")

public class AccountController {

@GetMapping("/details")

public Map<String, Object> getAccountDetails() {

return Map.of(

"accountNumber", "1234567890",

"accountHolder", "Jane Doe",

"balance", 5000.75,

"status", "ACTIVE"

);

}

}

**pom.xml:**

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>account</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>account-service</name>

<description>Account Service with Eureka Client</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**application.properties**:

server.port=8081

spring.application.name=ACCOUNT-SERVICE

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

eureka.instance.prefer-ip-address=true

management.endpoints.web.exposure.include=\*

management.endpoint.health.show-details=always

**AccountApplication.java**:

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

@EnableEurekaClient

public class AccountApplication {

public static void main(String[] args) {

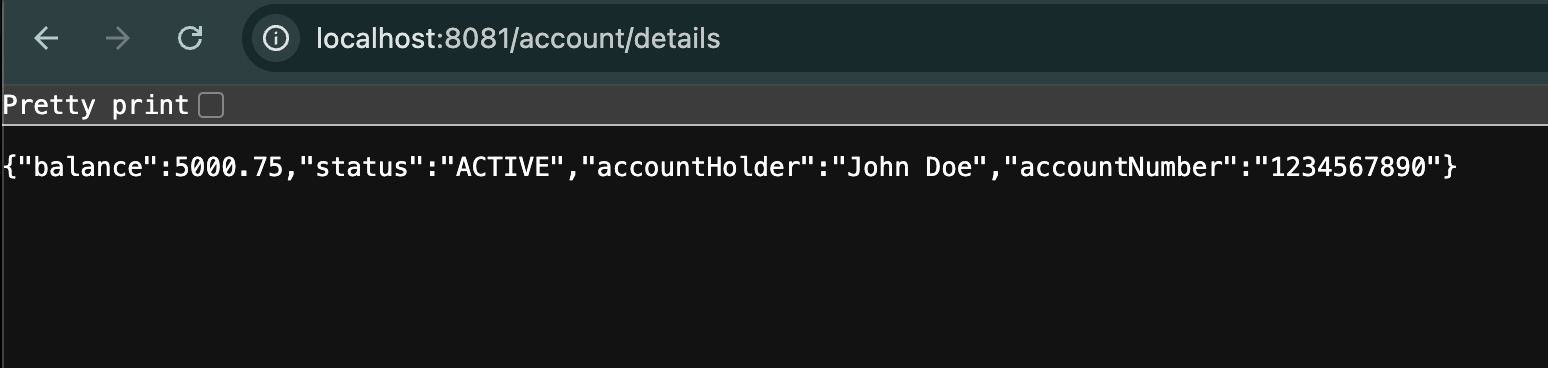
SpringApplication.run(AccountApplication.class, args);

}

}

**Step 6:**

Run **AccountApplication.java** as a java application and run <http://localhost:8081/account/details> on your browser.



**Step 7:**

Create a spring boot project named account.Refer below details

* + **Project:** Maven
  + **Language:** Java
  + **Spring Boot:** 3.2.4
  + **Group:** com.cognizant
  + **Artifact:** loan
  + **Name:** loan
  + **Package Name:** com.cognizant.loan
  + **Packaging:** Jar
  + **Java version:** 17

**Dependencies:**

* + Spring Boot DevTools
  + Spring Web
  + Eureka Discovery Client

**Step 8:**

Create the package named controller and also create class named **LoanController.java**.And alter **application.properties** and **LoanApplication.java.**Finally ensure the **pom.xml** too.

**pom.xml**:

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.2.4</version>

<relativePath/>

</parent>

<groupId>com.cognizant</groupId>

<artifactId>loan</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>loan-service</name>

<description>Loan Service with Eureka Client</description>

<properties>

<java.version>17</java.version>

<spring-cloud.version>2023.0.1</spring-cloud.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-netflix-eureka-client</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-dependencies</artifactId>

<version>${spring-cloud.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**LoanController.java**:

package com.cognizant.loan.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import java.util.Map;

@RestController

@RequestMapping("/loan")

public class LoanController {

@GetMapping("/details")

public Map<String, Object> getLoanDetails() {

return Map.of(

"loanId", "LN12345678",

"borrowerName", "John Smith",

"amount", 500000.0,

"interestRate", 7.5,

"status", "APPROVED"

);

}

}

**LoanApplication.java**:

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

@SpringBootApplication

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.run(LoanApplication.class, args);

}

}

**application.properties**:

server.port=8082

spring.application.name=LOAN-SERVICE

eureka.client.service-url.defaultZone=http://localhost:8761/eureka/

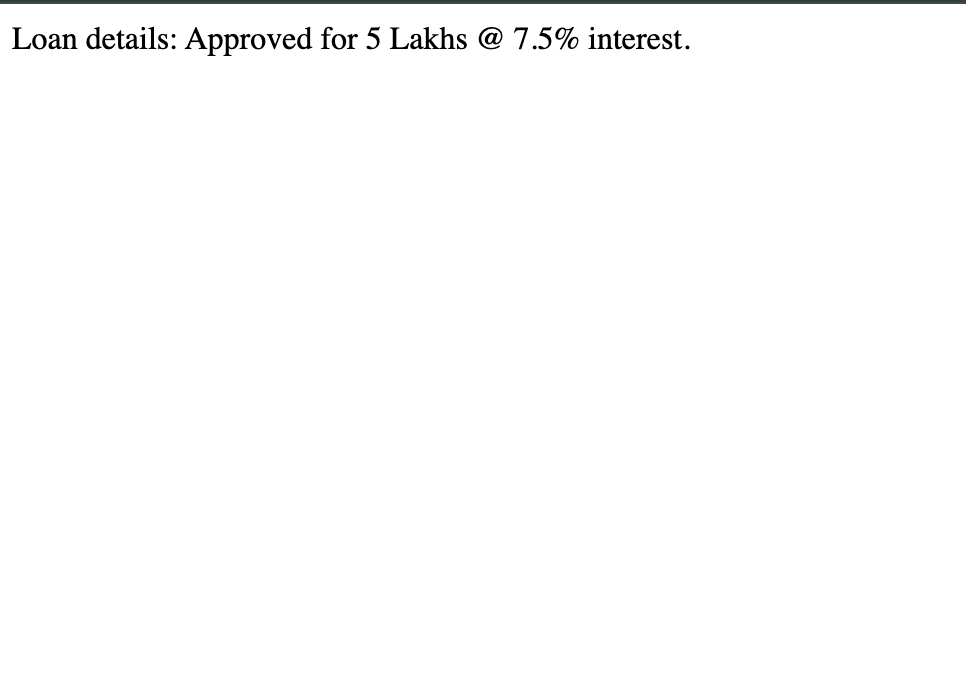
eureka.instance.prefer-ip-address=true

management.endpoints.web.exposure.include=\*

management.endpoint.health.show-details=always

**Step 9:**

Run **LoanApplication.java** as a java application and run http://localhost:8082/loan/details on your browser.



And then visit <http://localhost:8761/> on your browser to see the instance status.**Note:**The server of eurekaserver,account and loan application should start.

