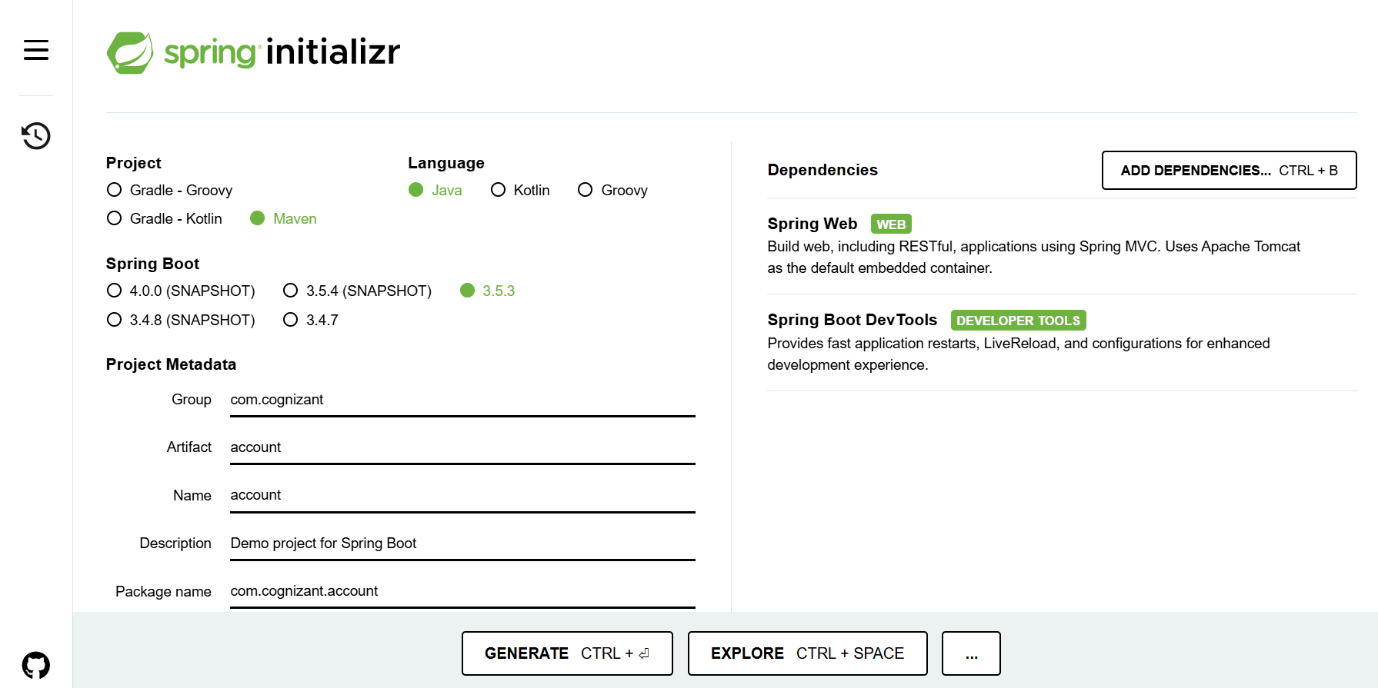
**Week4\_** **Microservices with API gateway\_HandsOn**

**Exercise 1: Creating Microservices for account and loan**

**Step 1 - Create Account Microservice:**

****

**Code:**

AccountController.java:

package com.cognizant.account.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/accounts")

public class AccountController {

@GetMapping("/{number}")

public Map<String, Object> getAccount(@PathVariable String number) {

return Map.of(

"number", number,

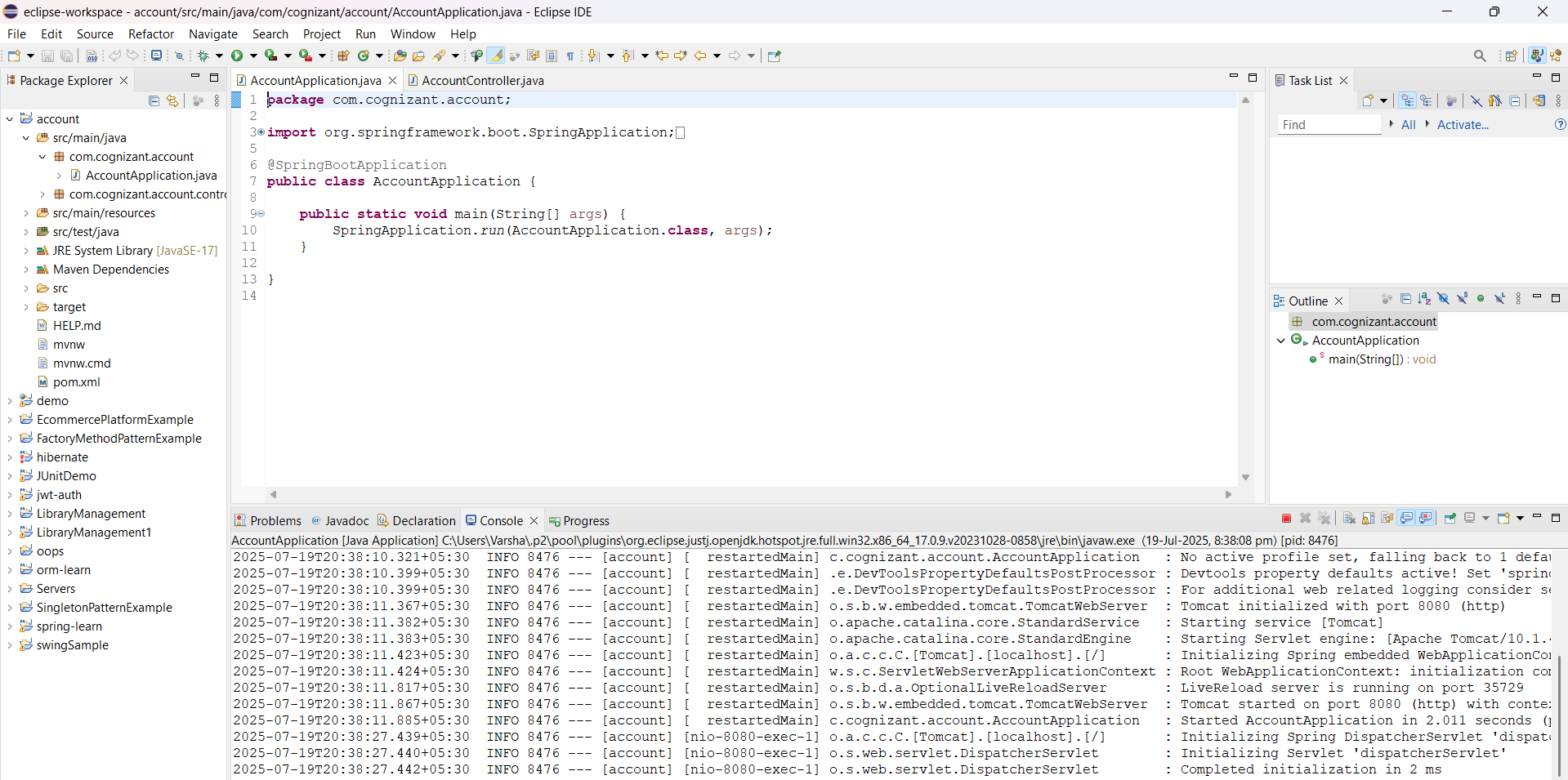
"type", "savings",

"balance", 234343

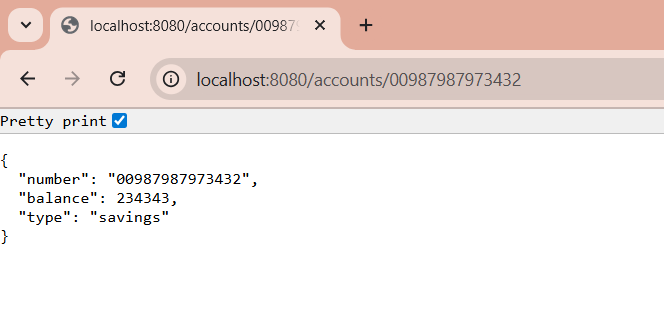
);

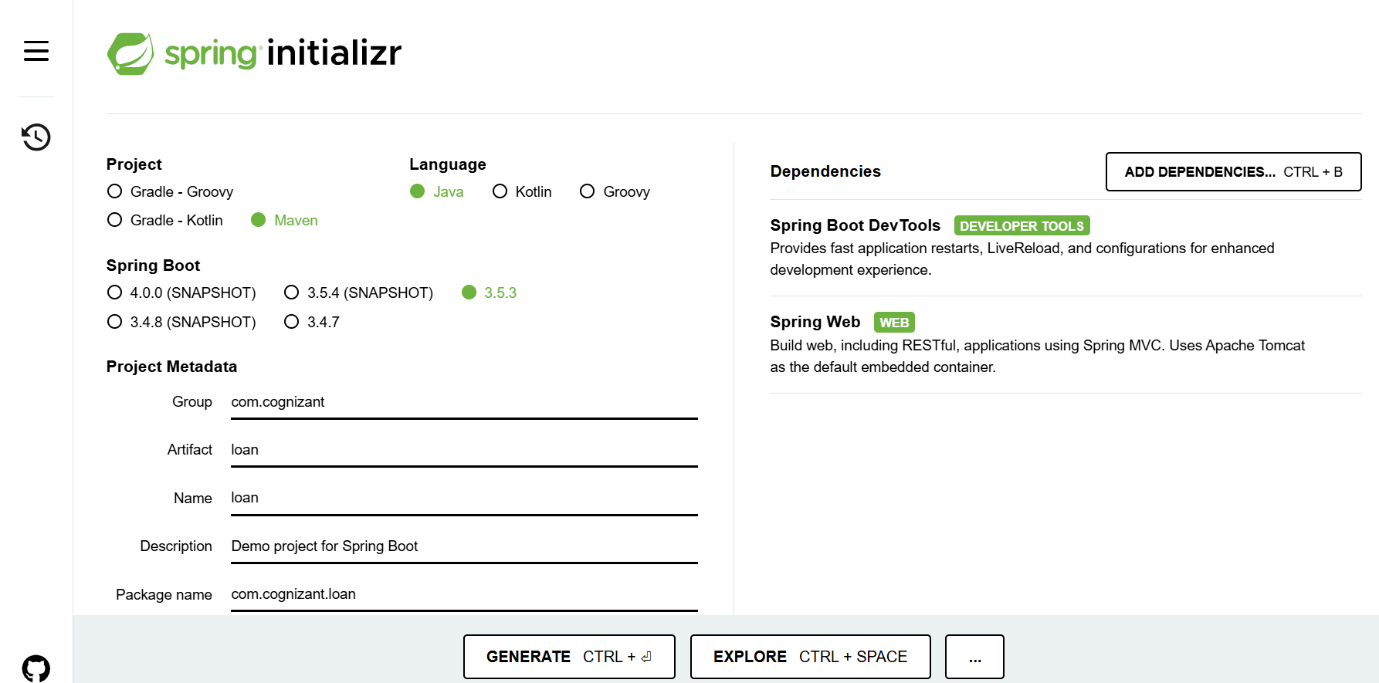
}

}

**Output:**

**Test in browser:**

http://localhost:8080/accounts/00987987973432

**Step 2 - Create Loan Microservice:**

**Code:**

LoanController.java:

package com.cognizant.loan.controller;

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

import java.util.Map;

@RestController

@RequestMapping("/loans")

public class LoanController {

@GetMapping("/{number}")

public Map<String, Object> getLoan(@PathVariable String number) {

return Map.of(

"number", number,

"type", "car",

"loan", 400000,

"emi", 3258,

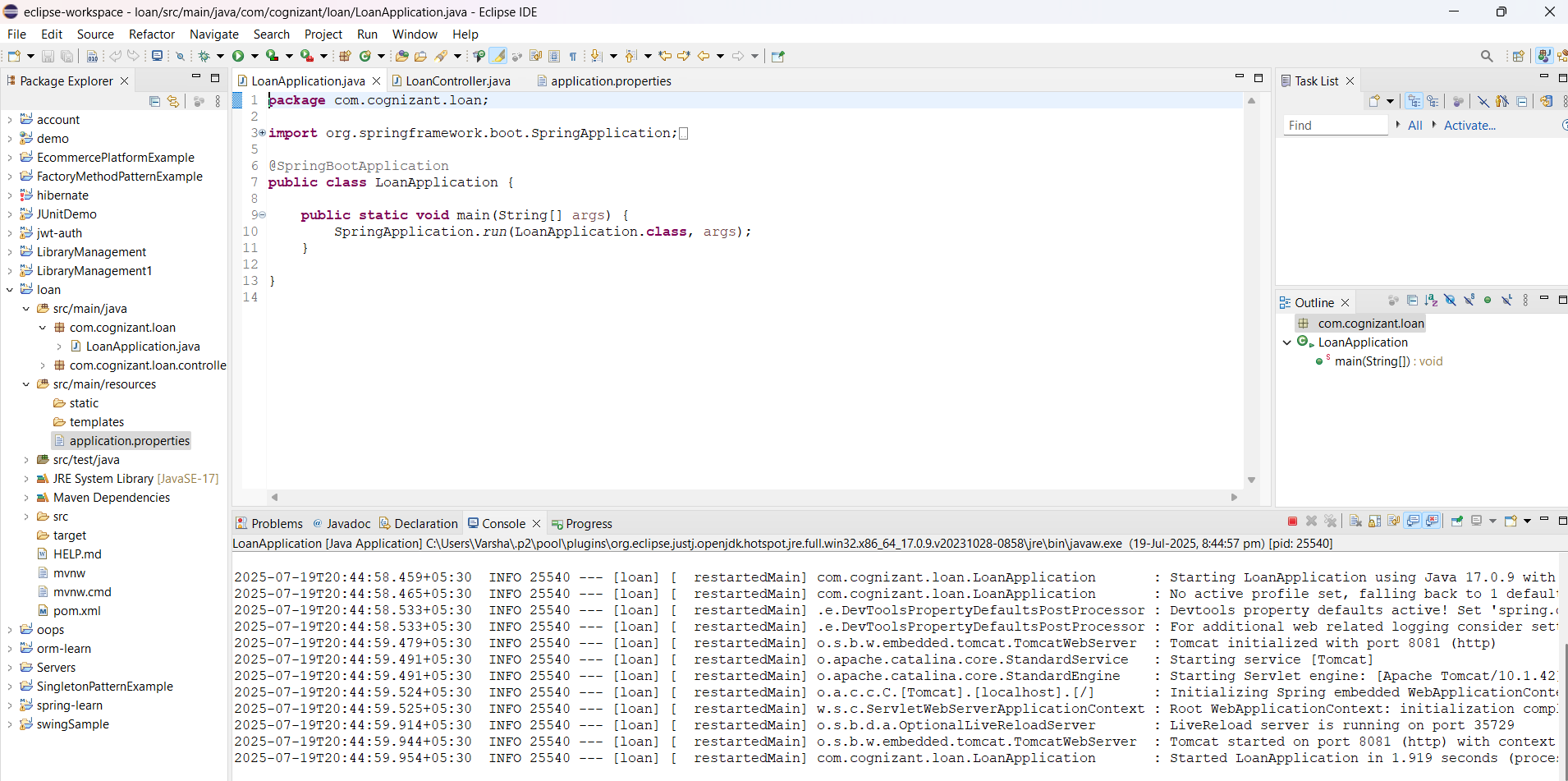
"tenure", 18

);

}

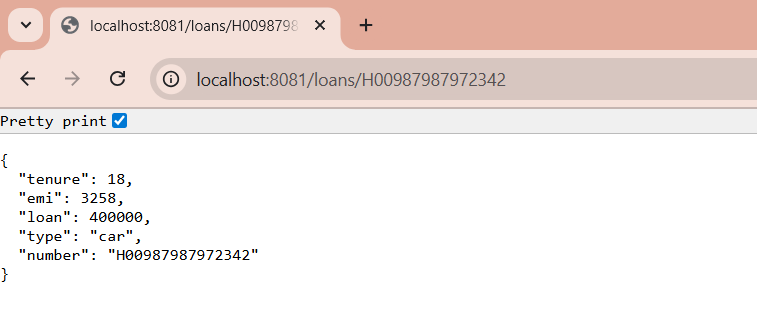
}

**Output:**

****

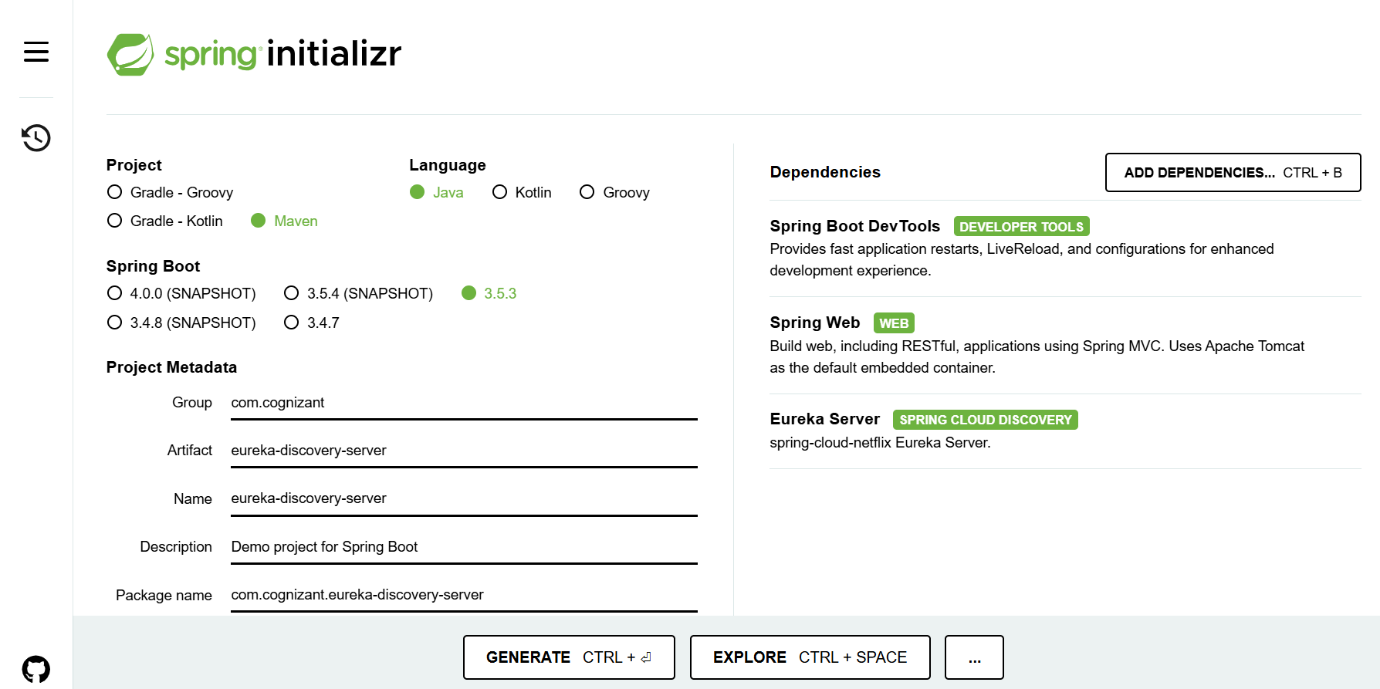
**Test in browser:**

http://localhost:8081/loans/H00987987972342

****

**Exercise 2: Create Eureka Discovery Server and register microservices**

**Step 1 - Create Eureka Discovery Server**

****

**Code:**

EurekaDiscoveryServerApplication.java:

package com.cognizant.eurekadiscoveryserver;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

@SpringBootApplication

@EnableEurekaServer

public class EurekaDiscoveryServerApplication {

public static void main(String[] args) {

SpringApplication.run(EurekaDiscoveryServerApplication.class, args);

}

}

application.properties:

server.port=8761

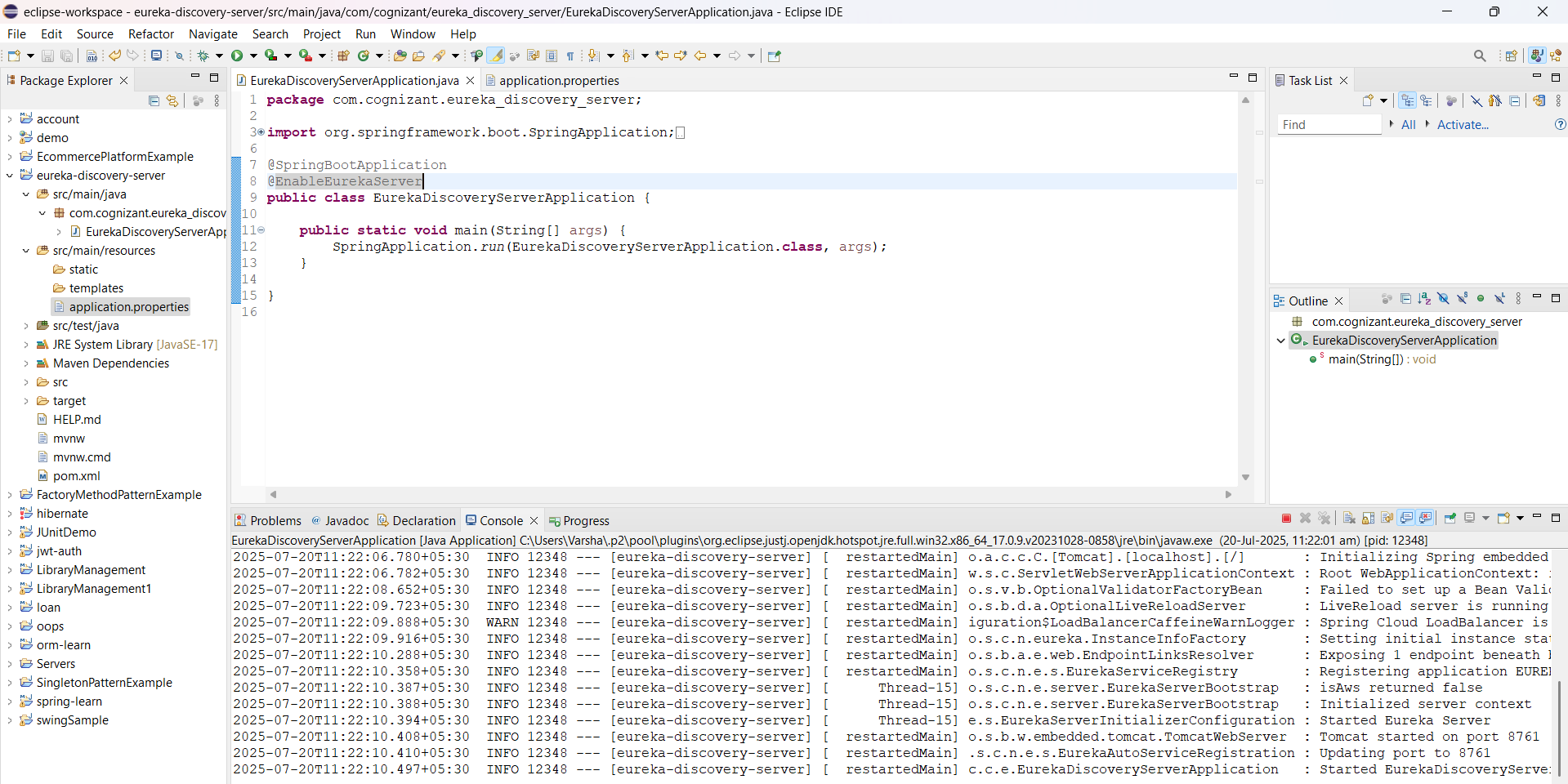
eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

logging.level.com.netflix.eureka=OFF

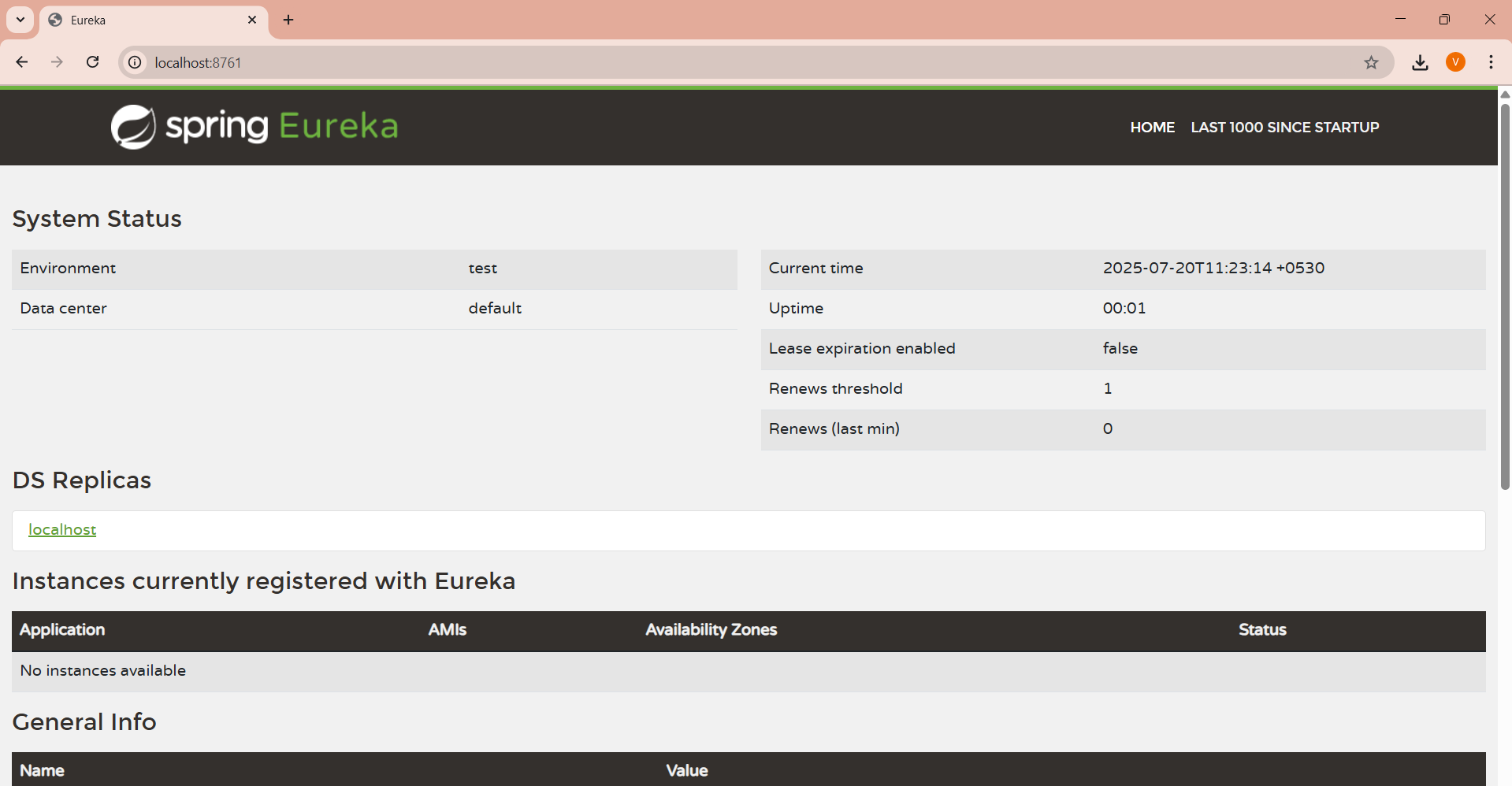
logging.level.com.netflix.discovery=OFF

**Output:**

****

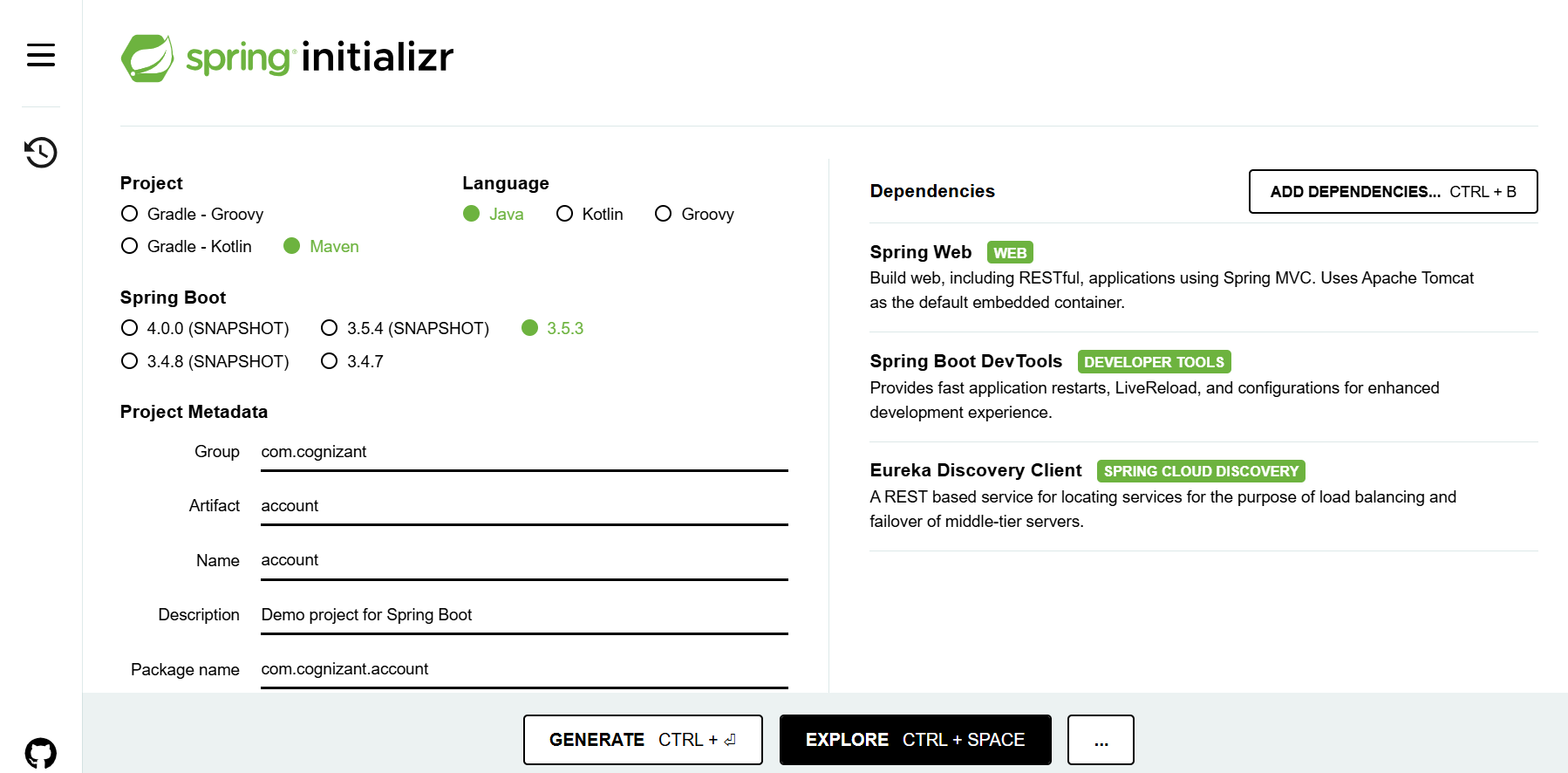
**Go to browser:**

http://localhost:8761

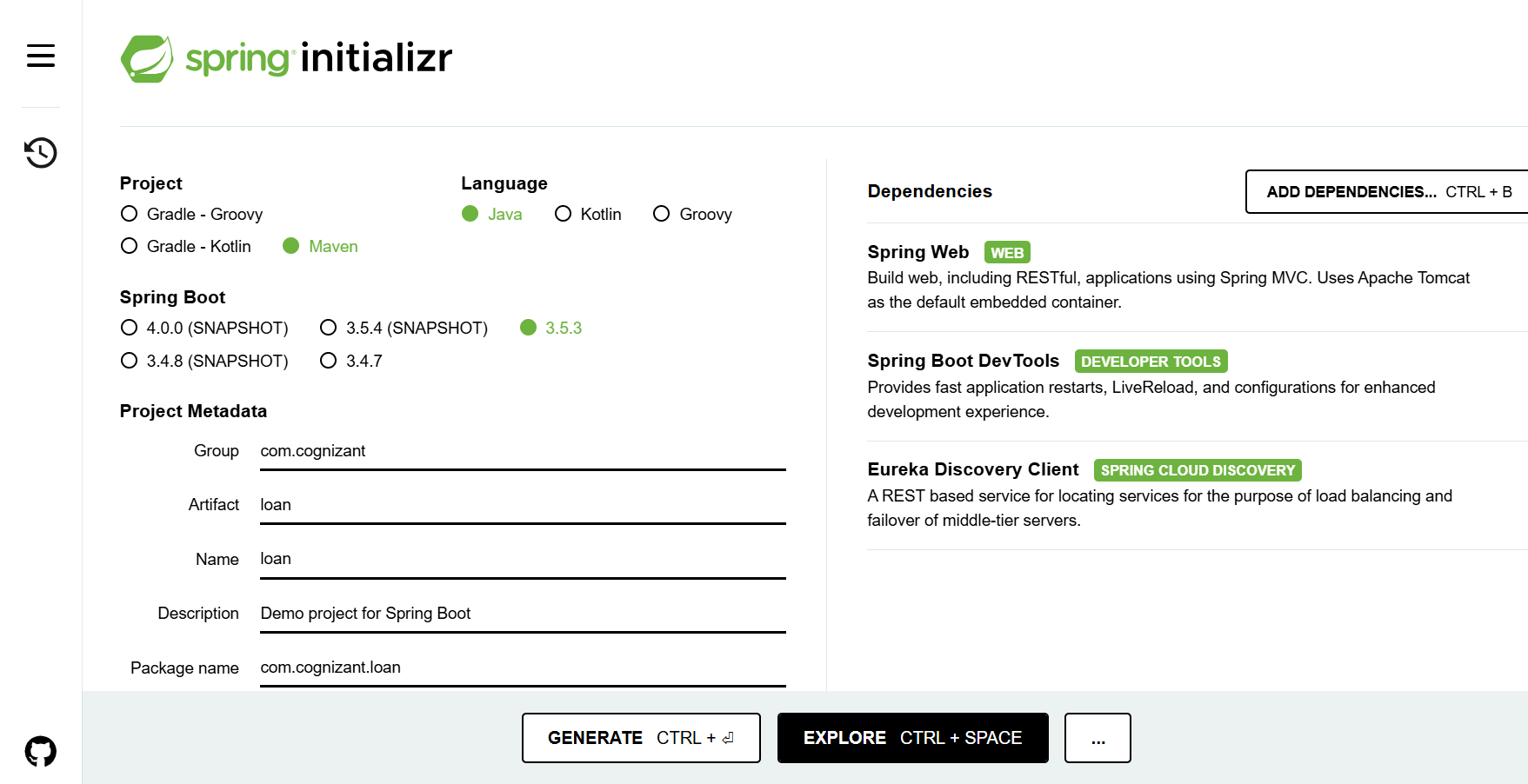
****

**Step 2 - Add Eureka Client Dependencies**

**Account:**

****

**Loan:**

****

**Step 3 - Enable Discovery Client in Application Class**

**Code:**

AccountApplication.java:

package com.cognizant.account;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class AccountApplication {

public static void main(String[] args) {

SpringApplication.*run*(AccountApplication.class, args);

}

}

LoanApplication.java:

package com.cognizant.loan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@SpringBootApplication

@EnableDiscoveryClient

public class LoanApplication {

public static void main(String[] args) {

SpringApplication.*run*(LoanApplication.class, args);

}

}

application.properties:

spring.application.name=account

server.port=8080

spring.application.name=account-service

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

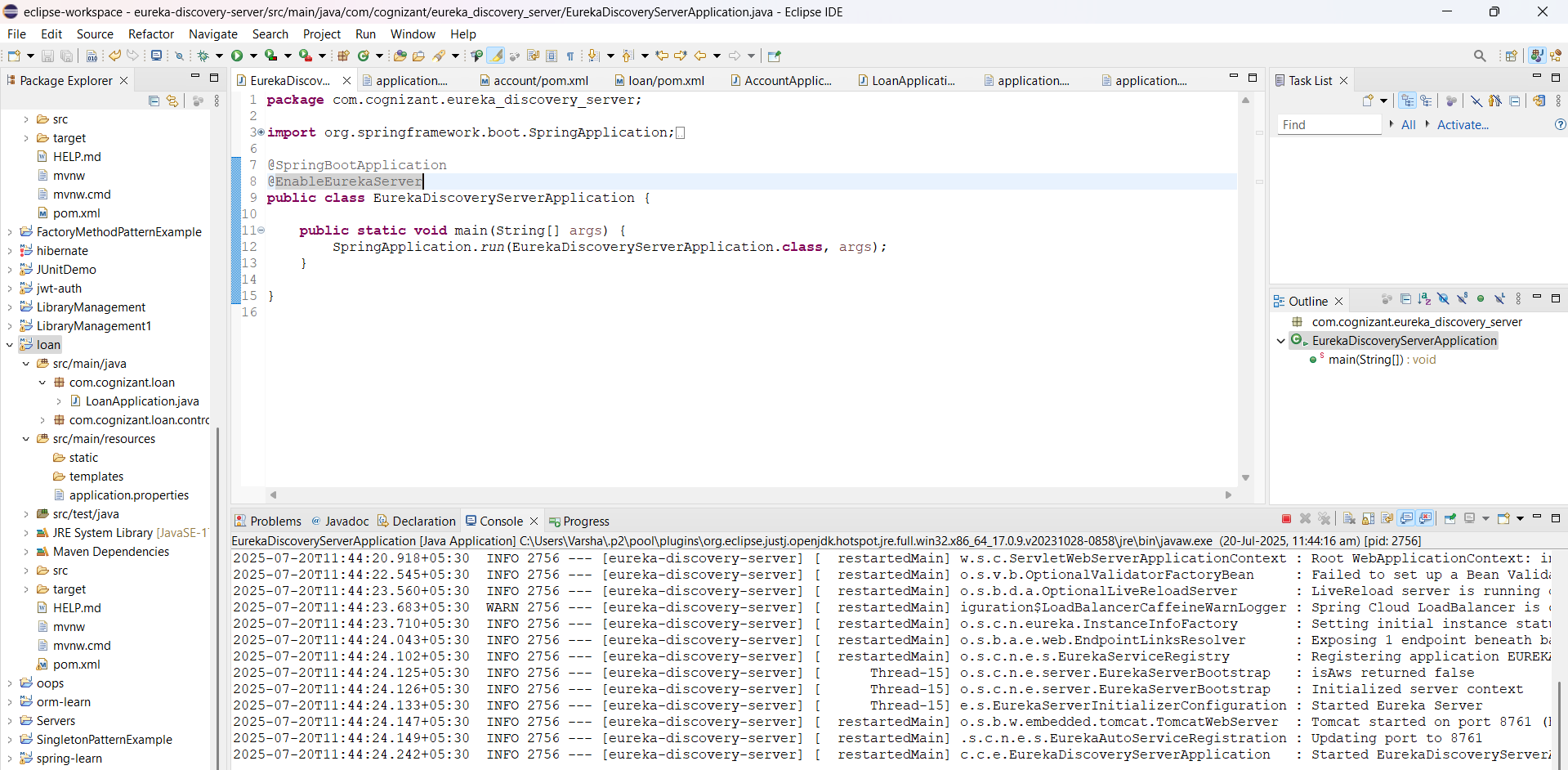
spring.application.name=loan

server.port=8081

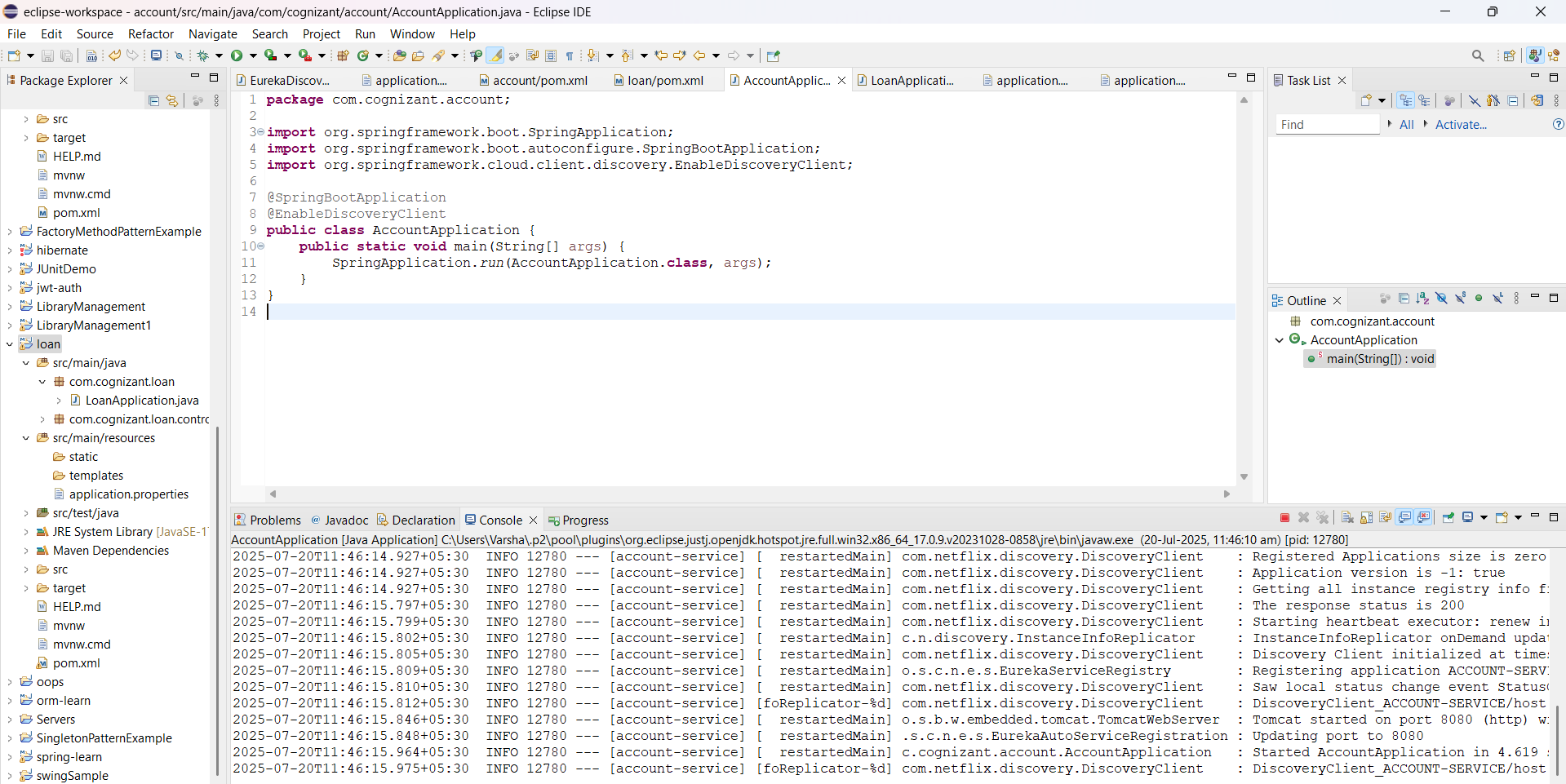
spring.application.name=loan-service

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

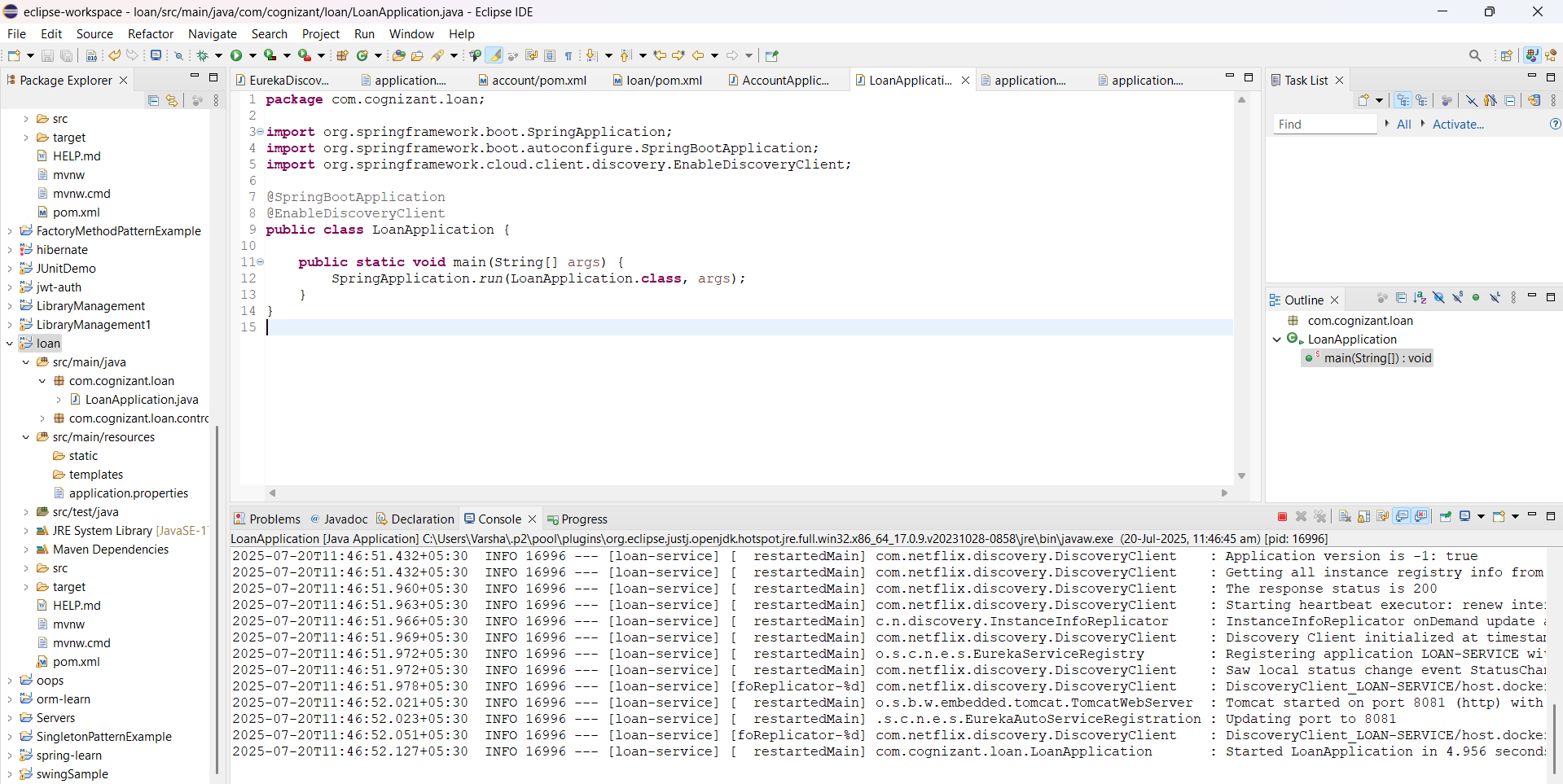
**Output:**

EurekaDiscoveryServerApplication.java:

AccountApplication.java:

****

LoanApplication.java:

****

**Go to browser:**

<http://localhost:8761>

****