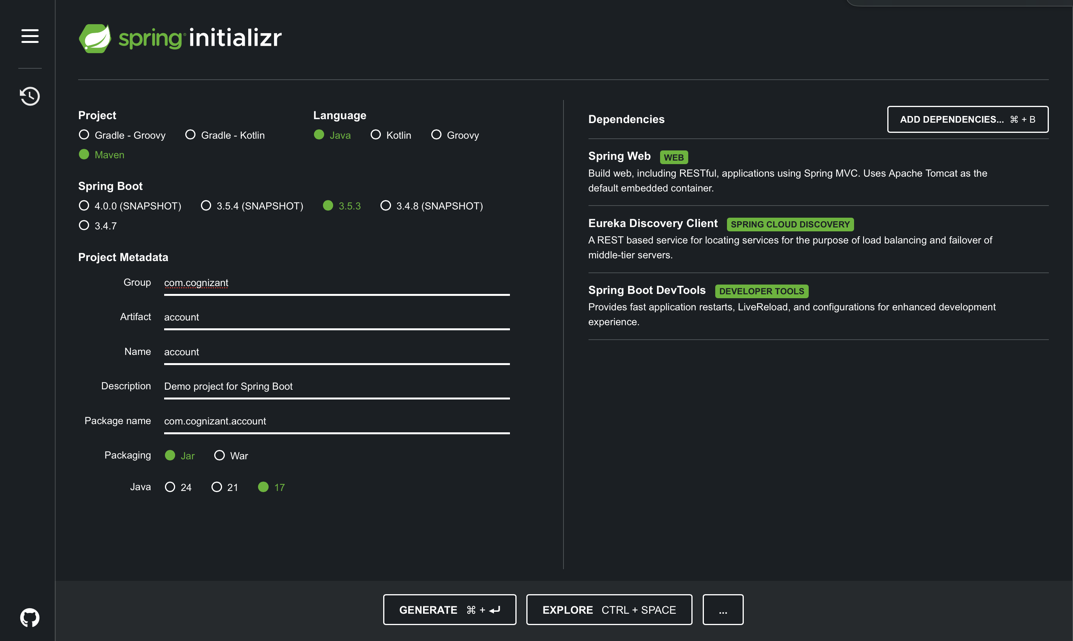
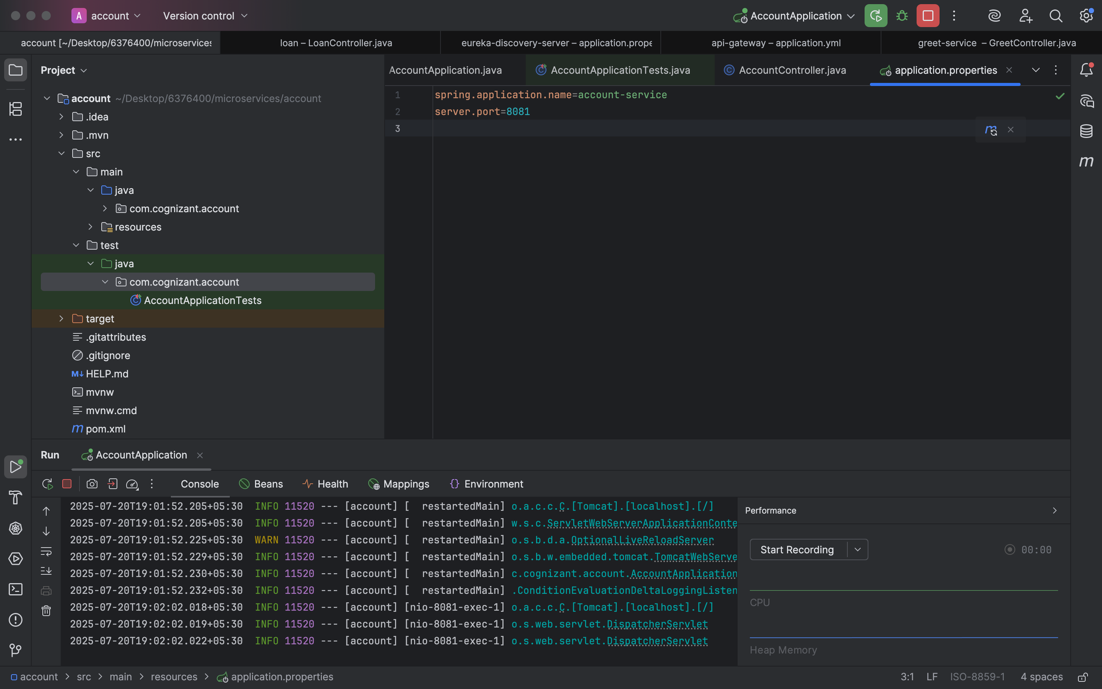
**Hands-on 1 : Creating Microservices for account and loan**

**A. Account Microservice**

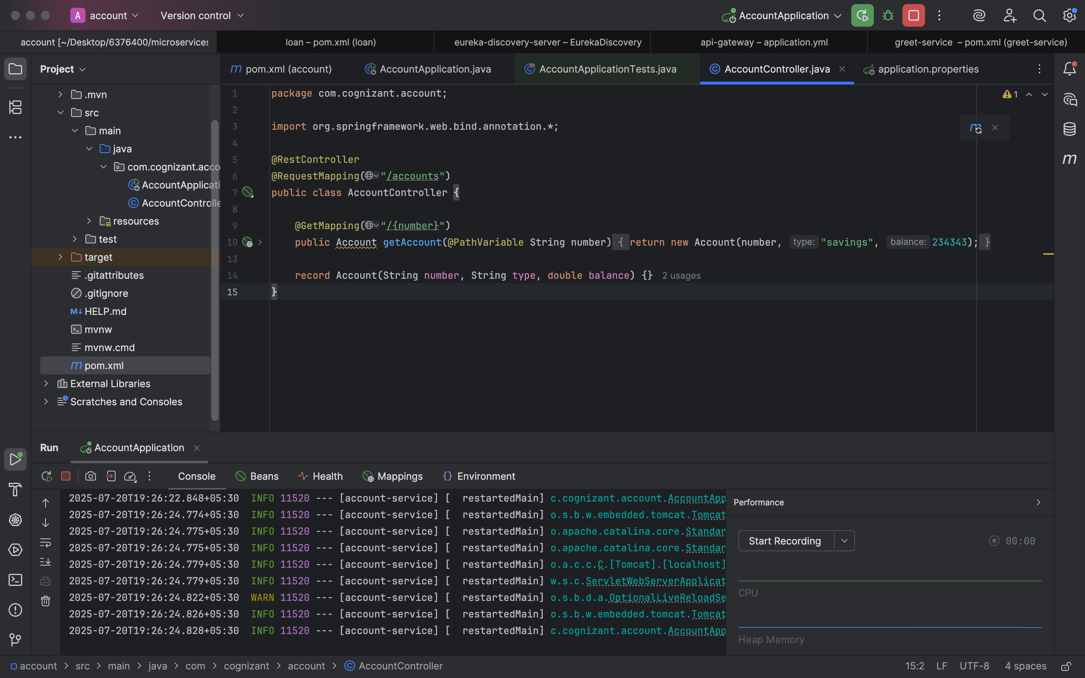
1. Go to https://start.spring.io and configure:
   * Group: com.cognizant
   * Artifact: account
   * Dependencies: Spring Web, Spring Boot DevTools, Eureka Discovery Client



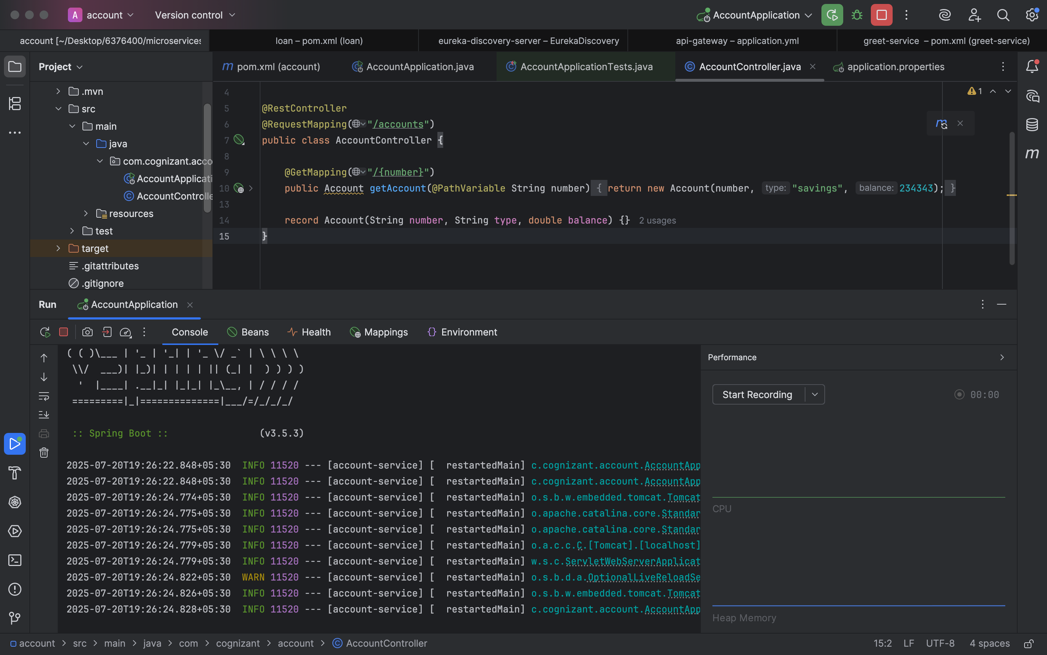
1. Extract ZIP and import into IntelliJ as Maven project.
2. Set  application.properties:

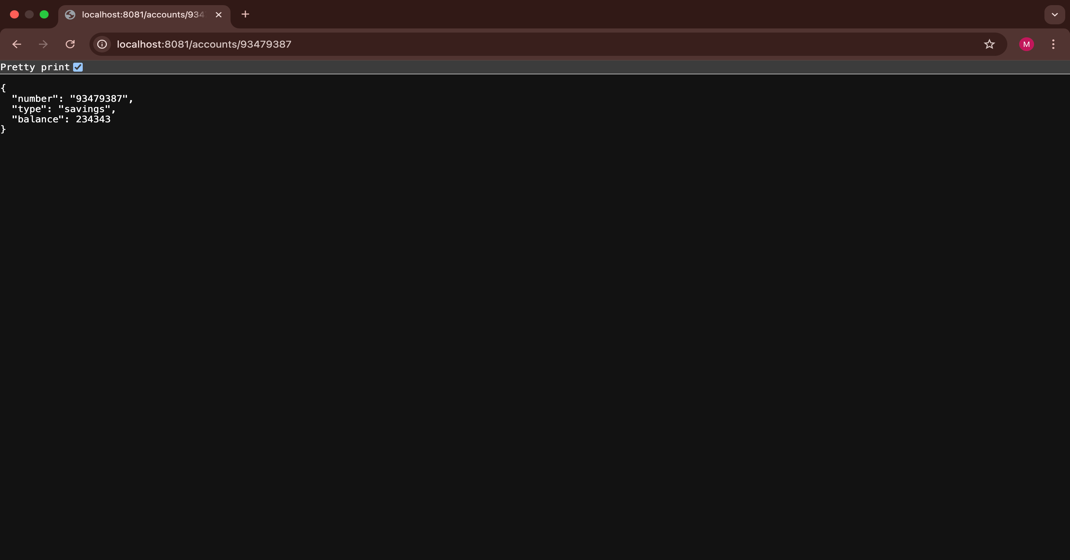


1. Create Controller with endpoint /account/{number}



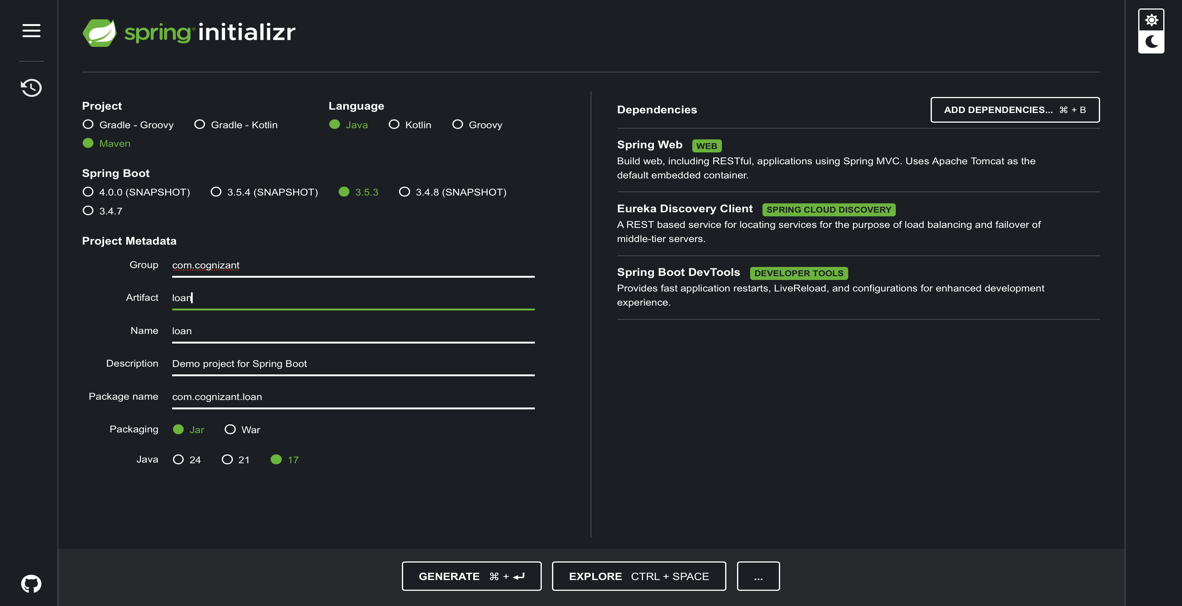
1. Run application and test http://localhost:8081/account/status



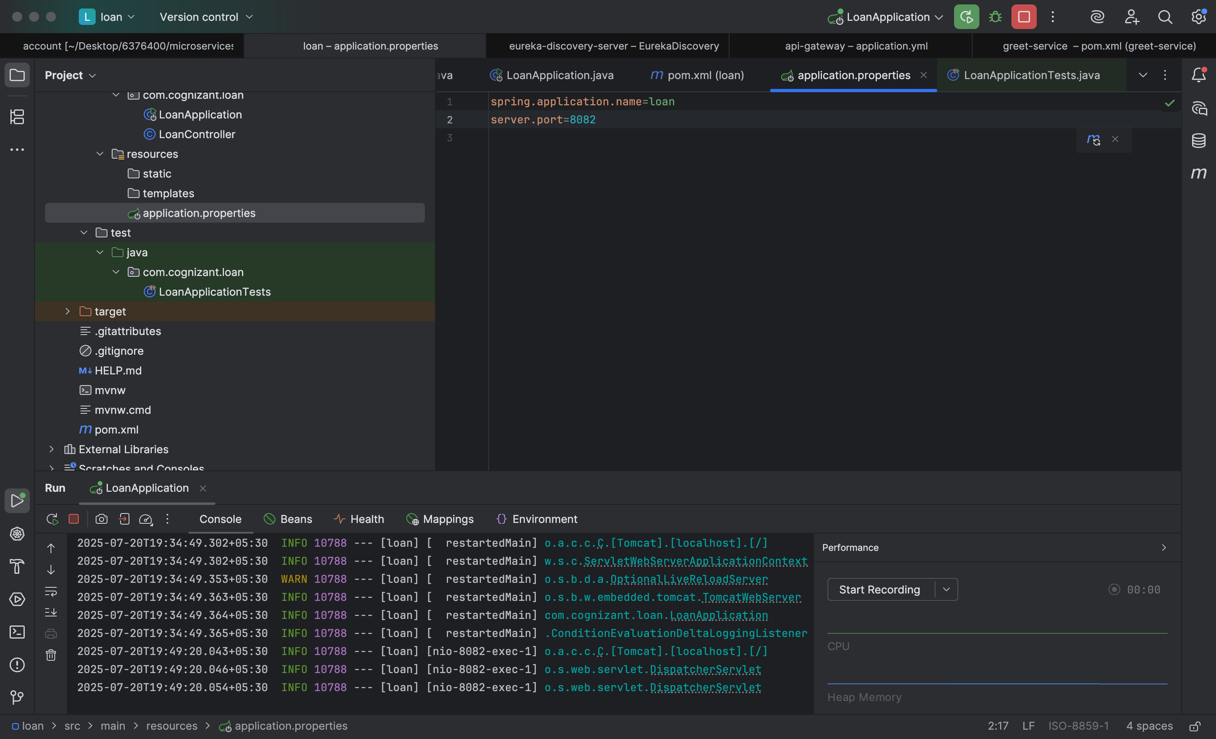


#### ****B. Loan Microservice****

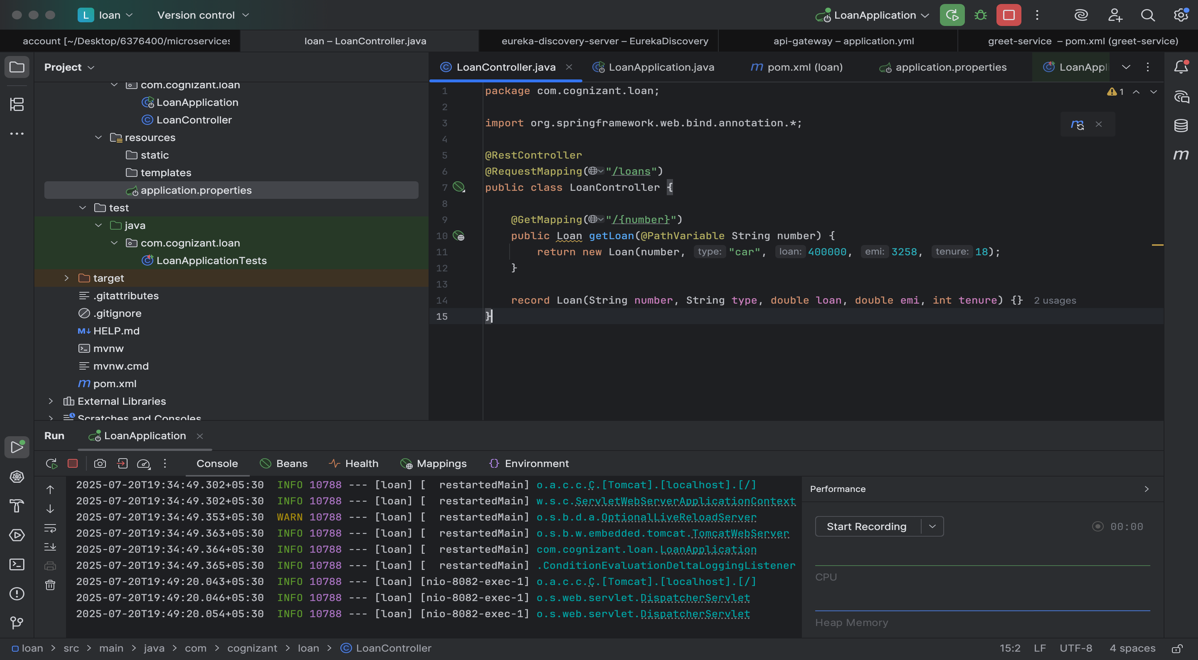
1. Go to https://start.spring.io and configure:
   * Group: com.cognizant
   * Artifact: loan
   * Dependencies: Spring Web, Spring Boot DevTools, Eureka Discovery Client



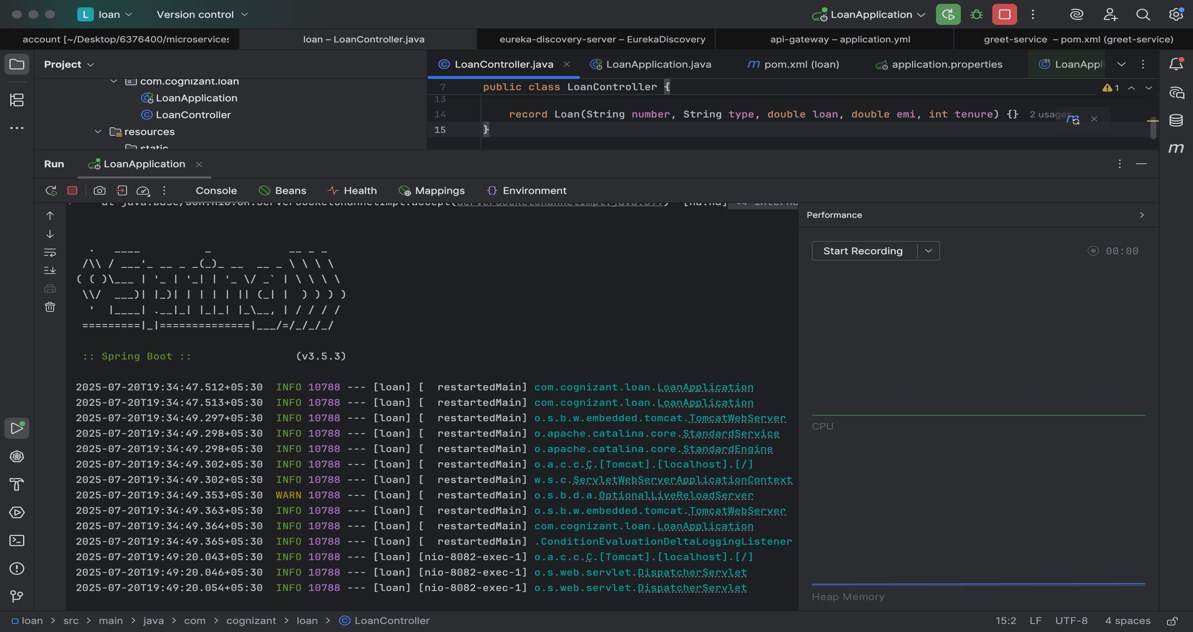
1. Extract ZIP and import into IntelliJ.
2. Set application.properties:

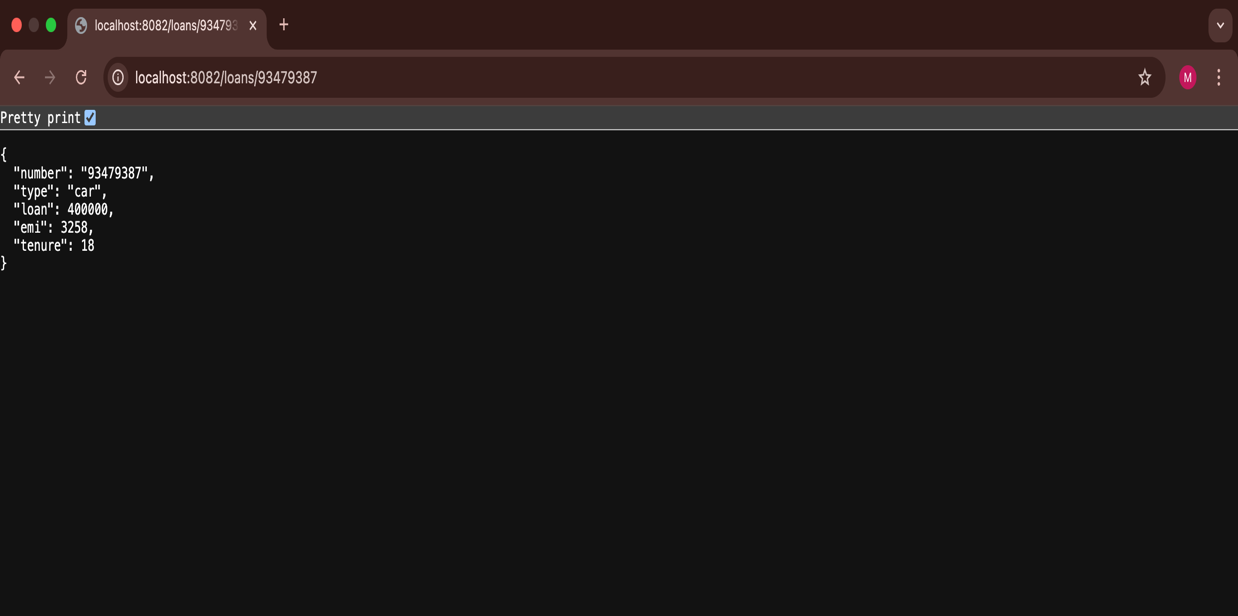


1. Create Controller with endpoint /loan/{number}



1. Run and test at <http://localhost:8082/loan/status>

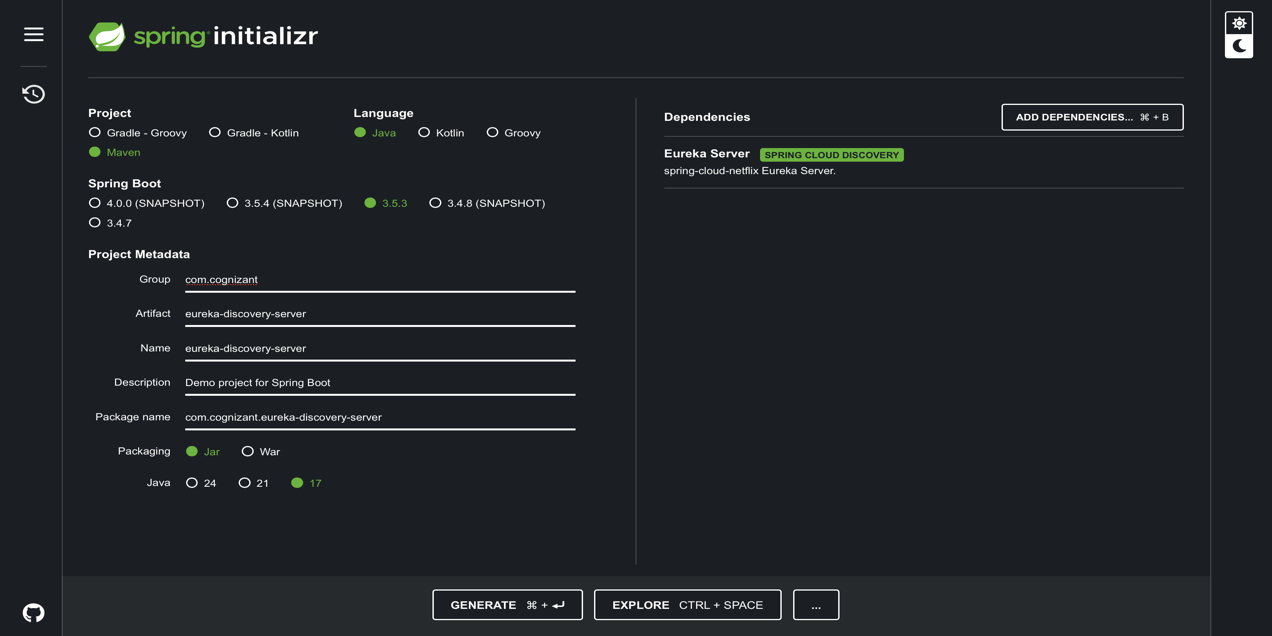




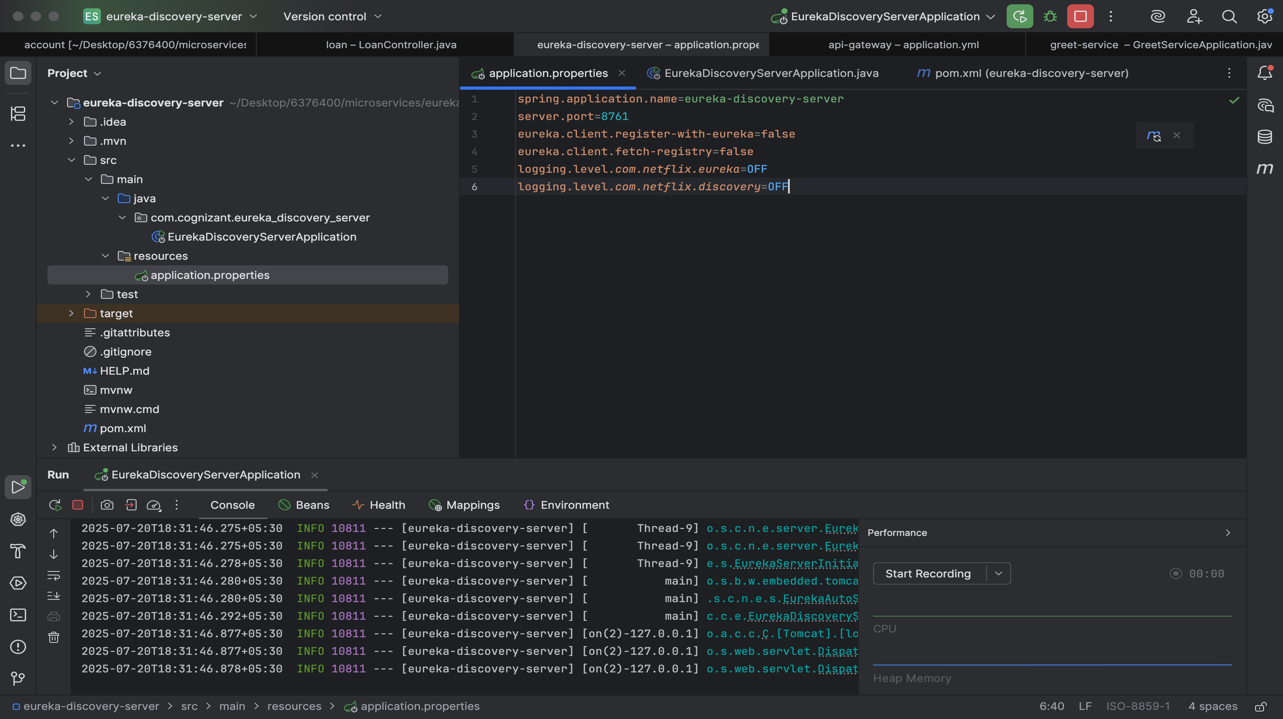
**Hands-on 1 : Create Eureka Discovery Server and register** **microservices:**

#### ****A. Eureka Discovery Server****

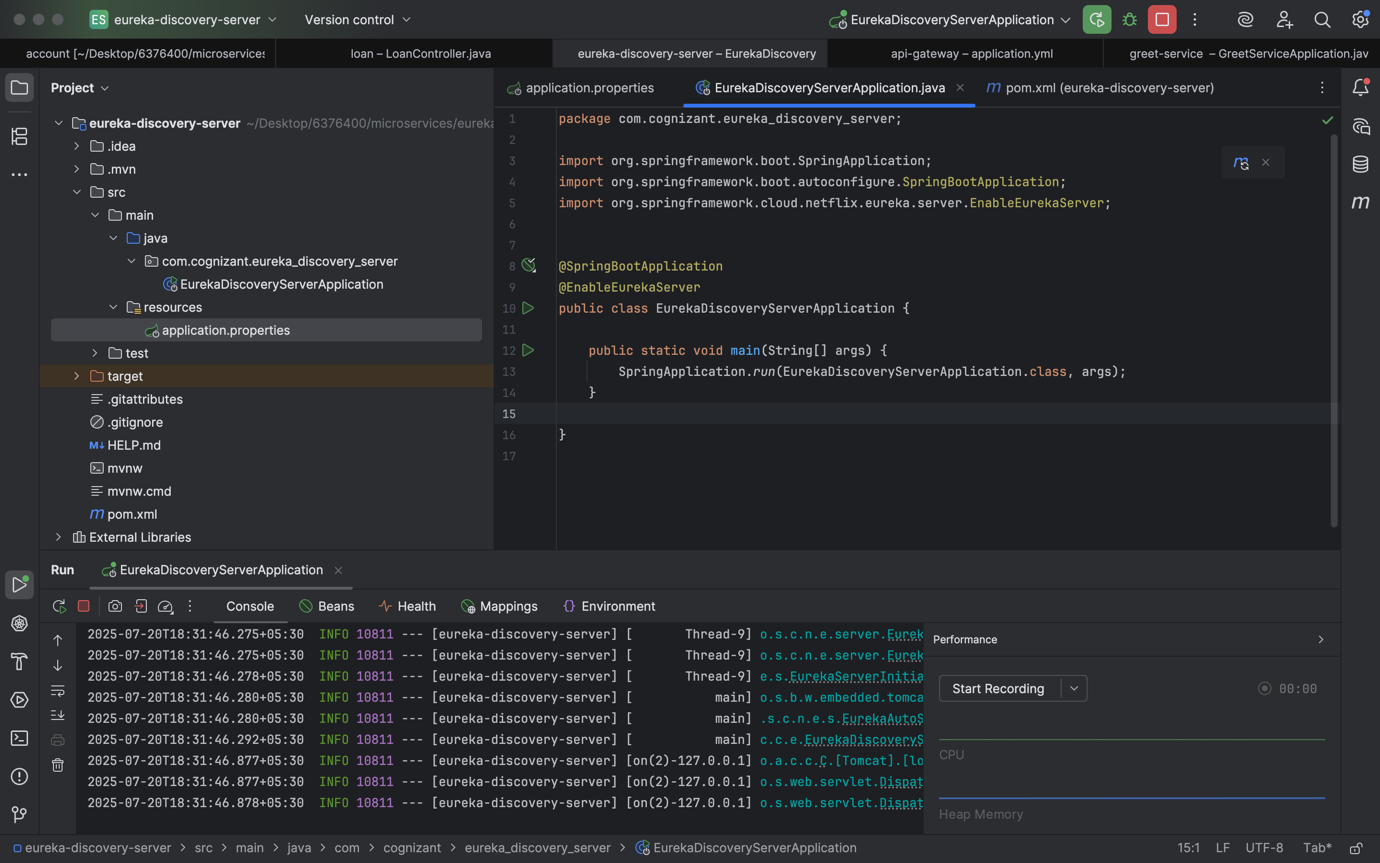
1. Go to https://start.spring.io:
   * Group: com.cognizant
   * Artifact: eureka-discovery-server
   * Dependencies: Eureka Server

****

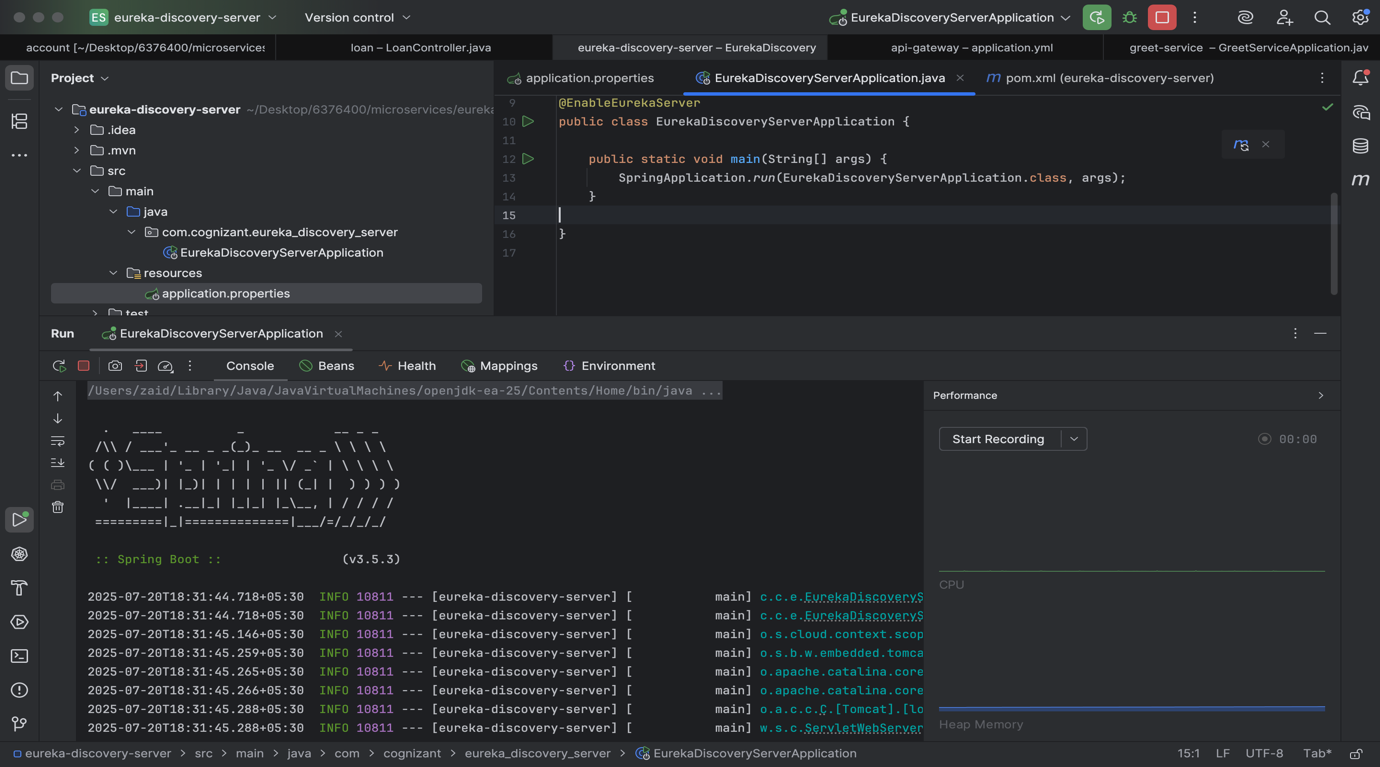
1. Extract and import into IntelliJ.
2. Set application.properties



1. Annotate main class with @EnableEurekaServer

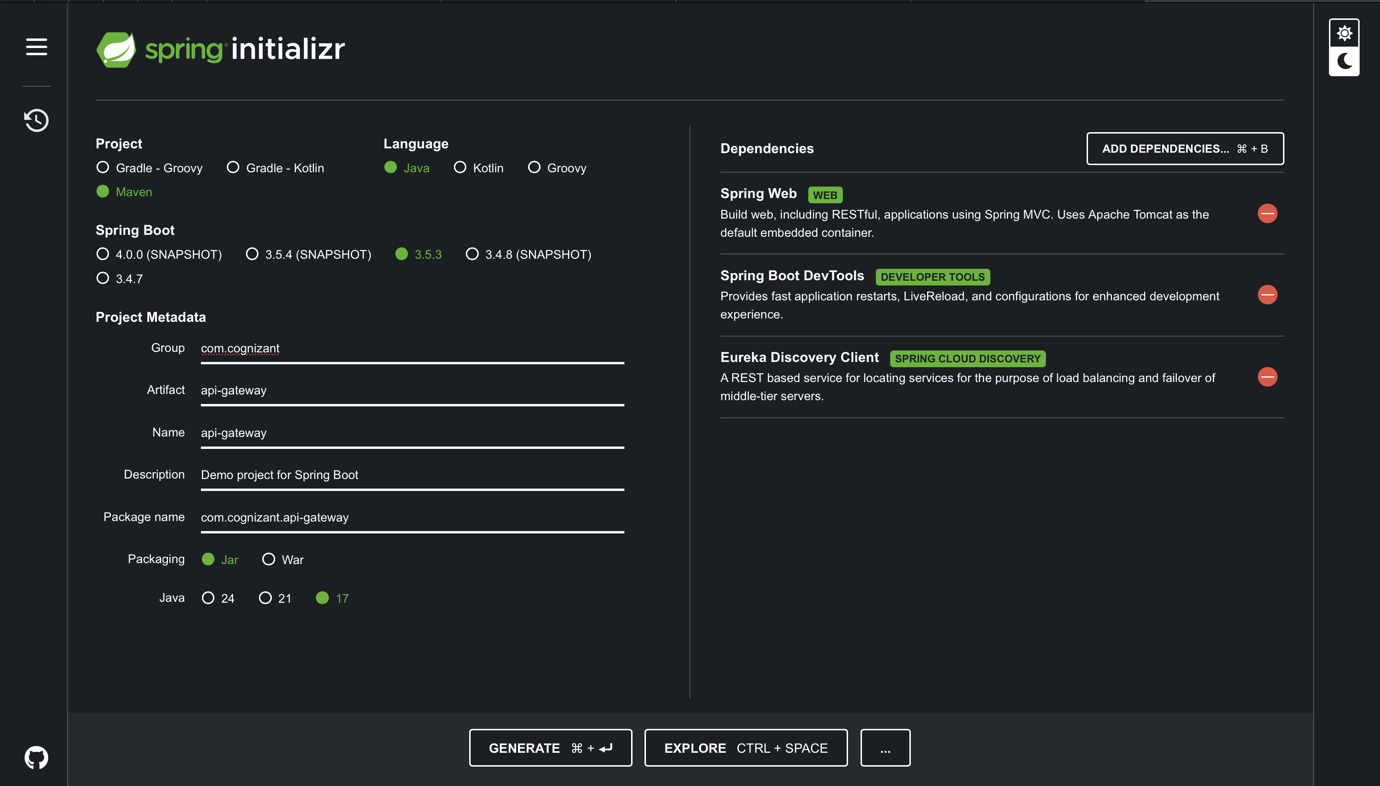


1. Run and visit http://localhost:8761 to confirm server is up

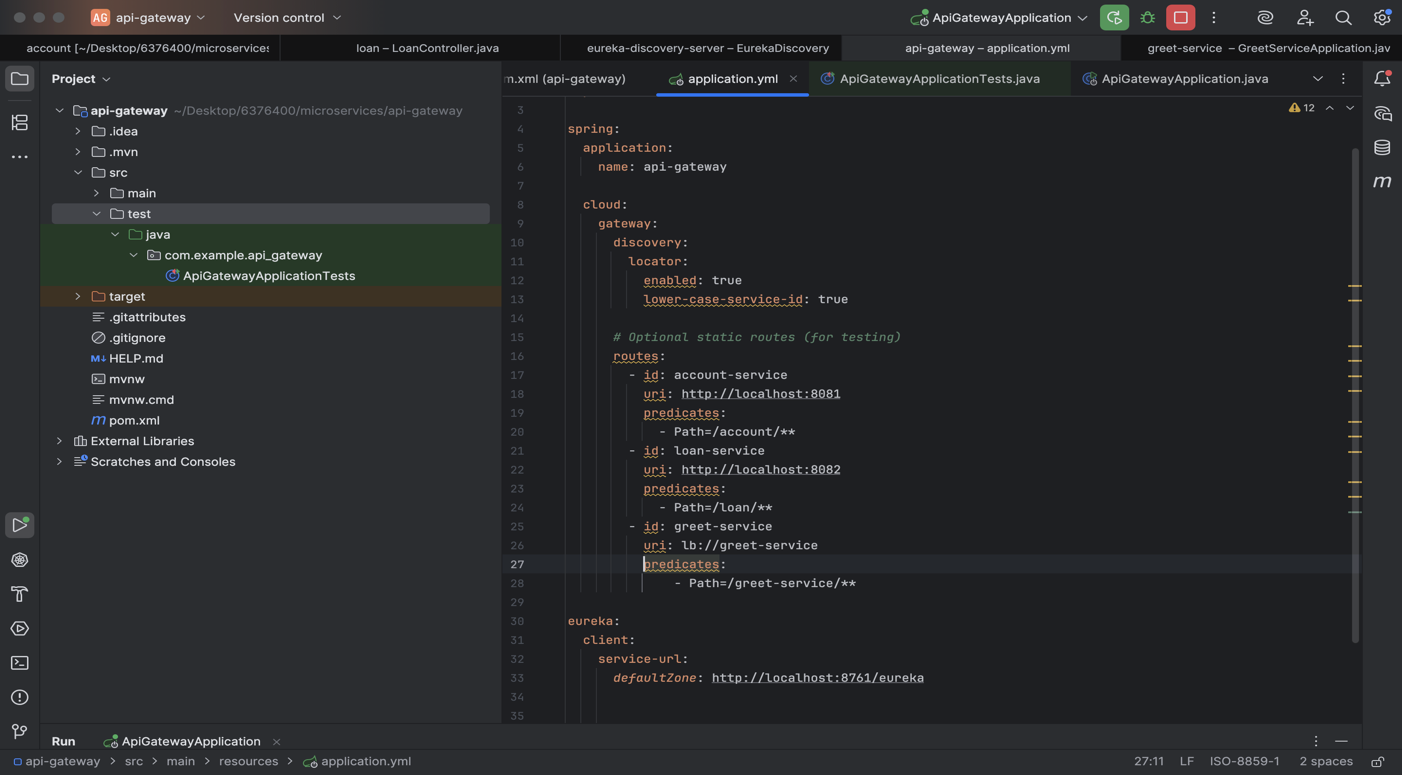


#### ****B. Spring Cloud API Gateway****

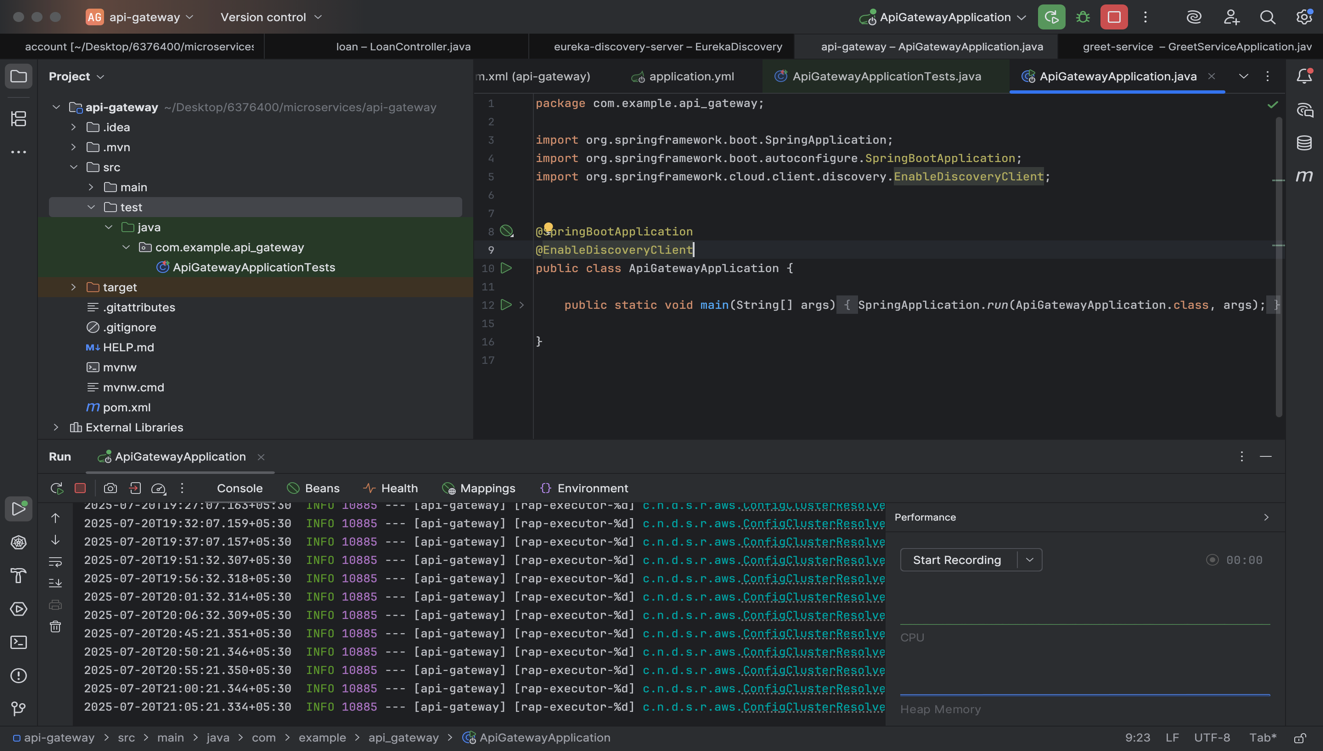
1. Go to https://start.spring.io:
   * Group: com.cognizant
   * Artifact: api-gateway
   * Dependencies: Spring Web, Spring Cloud Gateway, Eureka Discovery Client

****

1. Extract and import into IntelliJ.
2. Set application.yml:

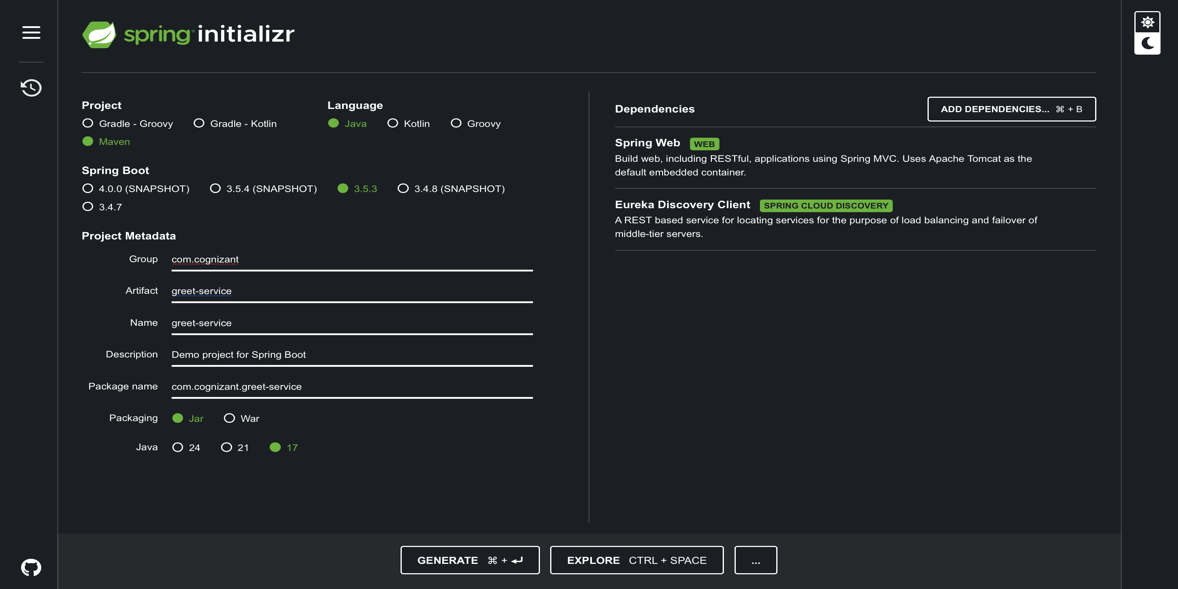


1. Annotate Main Class with @EnableDiscoveryClient

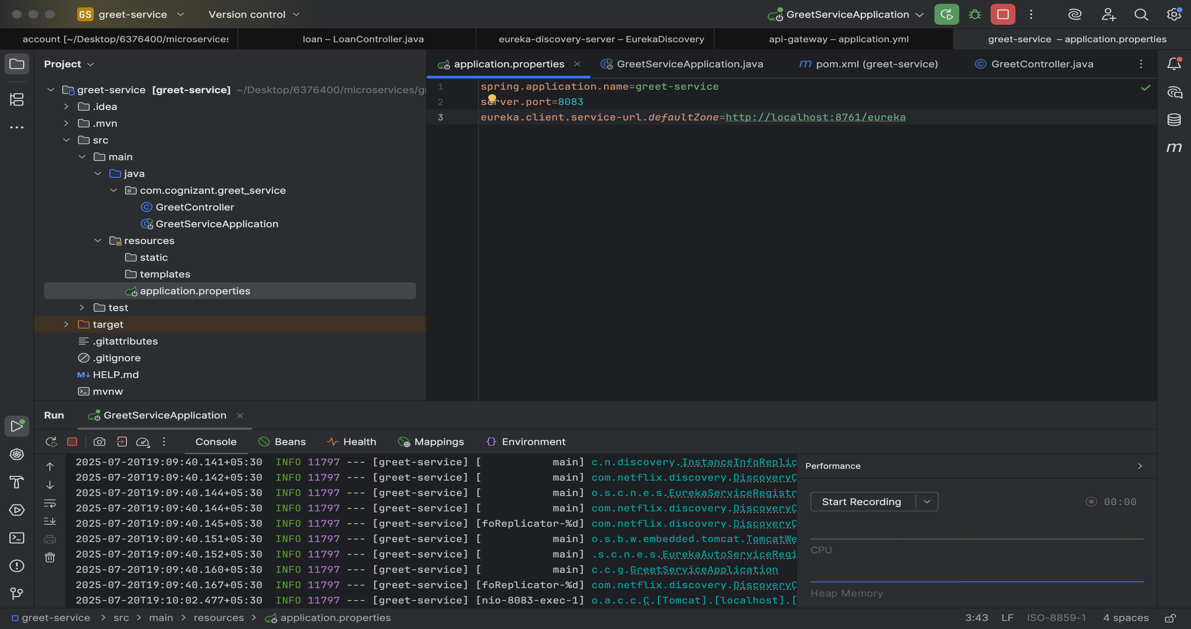


**C. Greet Microservice**

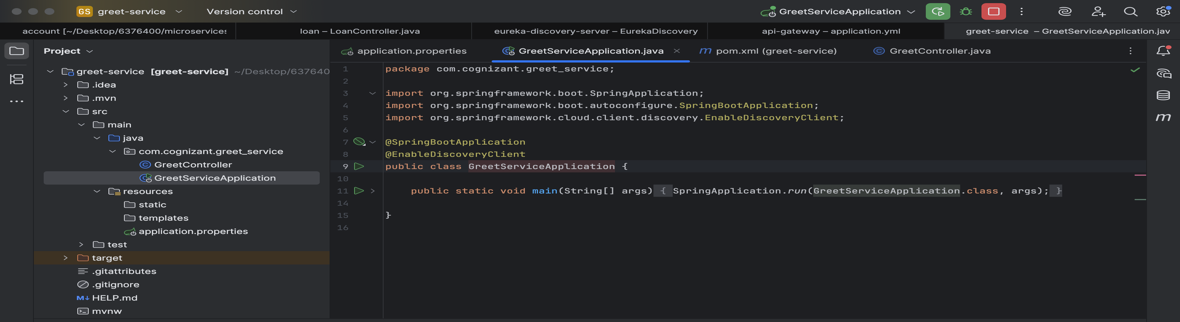
1. Go to https://start.spring.io:
   * Group: com.cognizant
   * Artifact: greet-service
   * Dependencies: Spring Web, Eureka Discovery Client



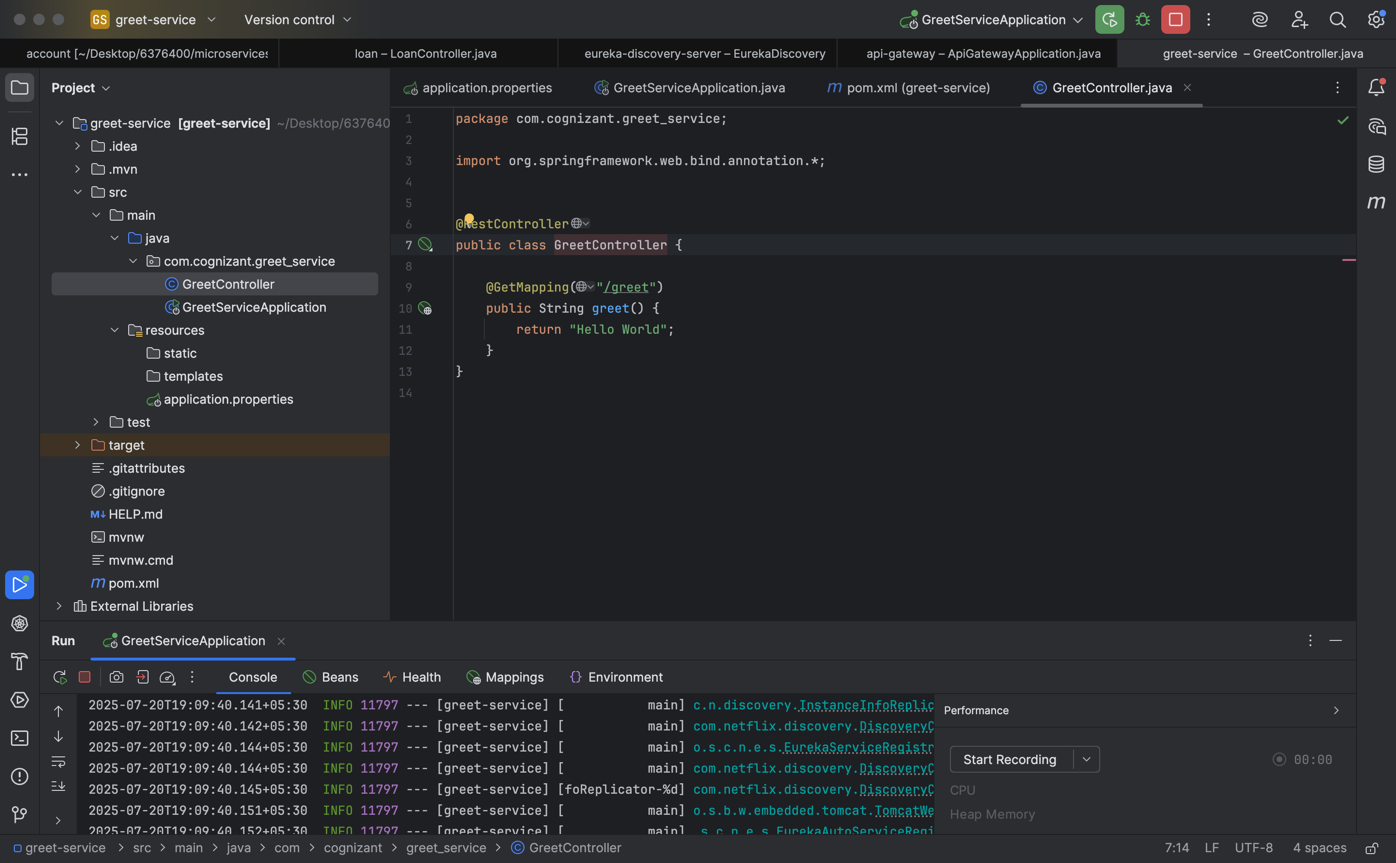
1. Extract ZIP and import to IntelliJ.
2. Set application.properties:



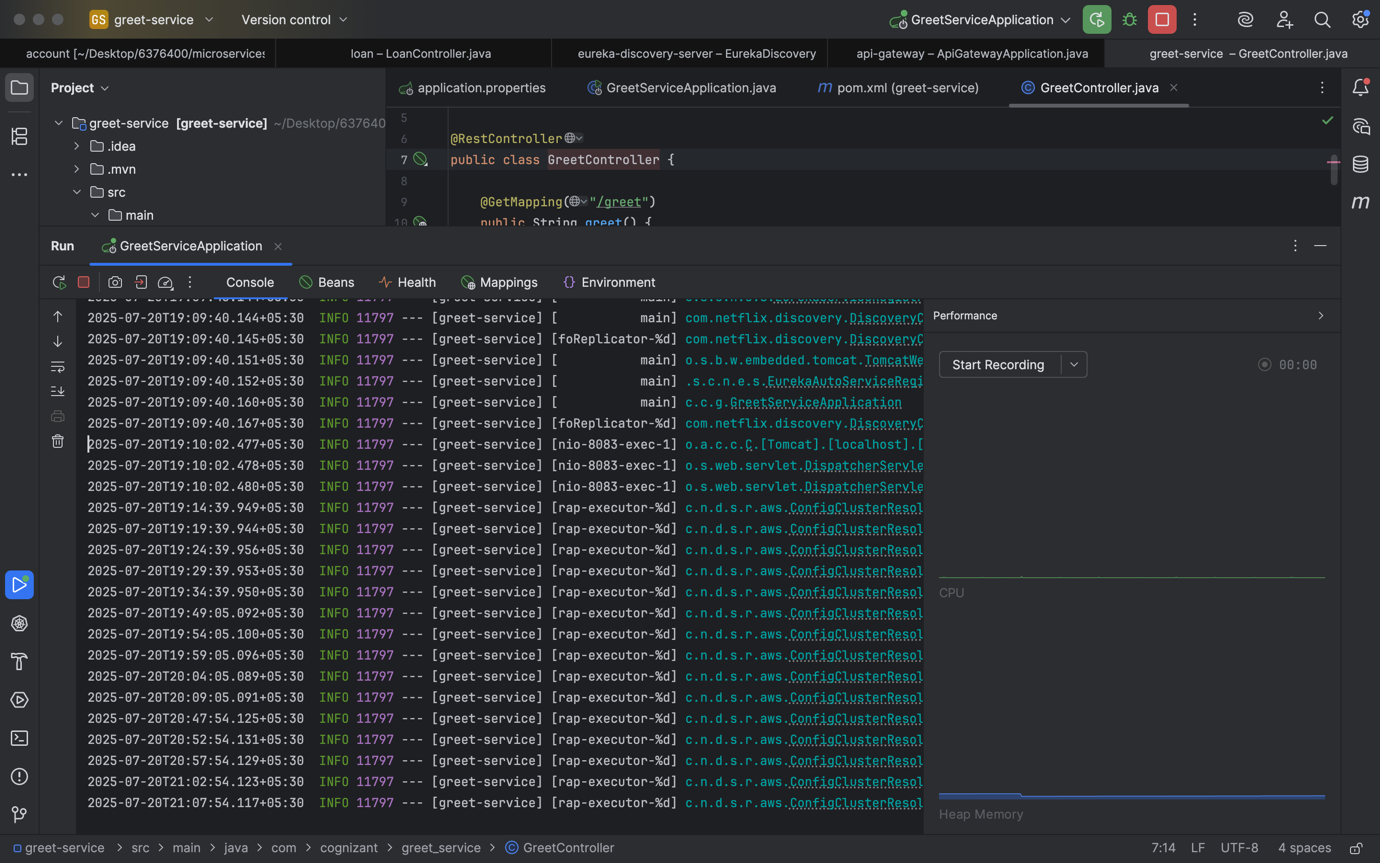
1. Create Main Class with @EnableDiscoveryClient

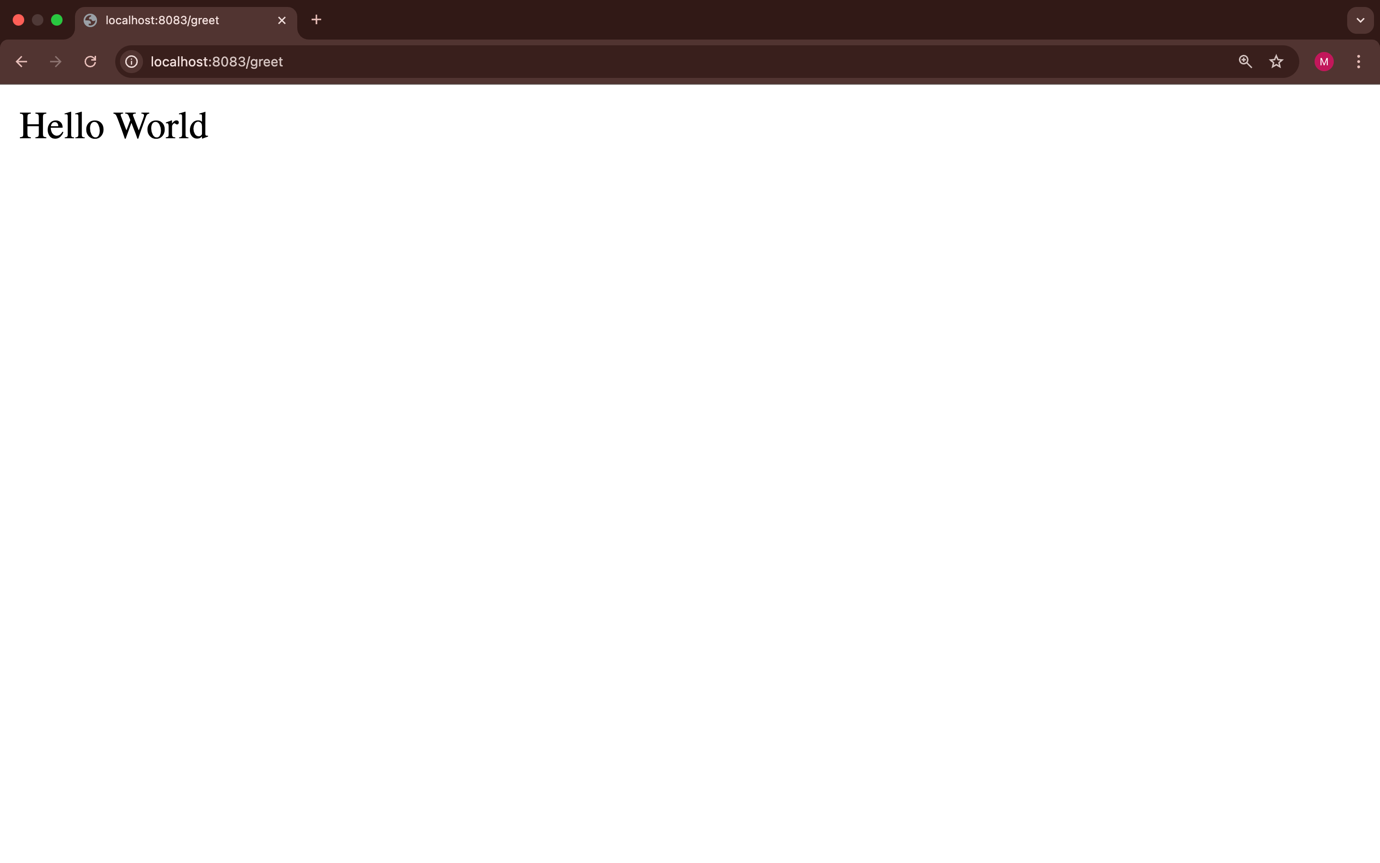


1. Create Controller with endpoint /greet returning "Hello World"



1. Run and test http://localhost:8083/greet





### **Concepts Explained:-**

#### ****Microservices Architecture:****

A software design pattern where each application function is developed and deployed independently as a service. It helps in scaling, managing, and updating parts of the application separately.

#### ****Spring Boot:****

A framework used to build stand-alone and production-ready Spring-based applications. It simplifies the configuration and development of Spring apps.

#### ****Eureka Discovery Server:****

A service registry provided by Netflix and integrated with Spring Cloud. Microservices register themselves to Eureka so they can be discovered by other services.

#### ****Spring Cloud API Gateway:****

A single entry point for all microservices. It handles routing, filtering, and load balancing. This avoids direct communication between client and each service.

#### ****Global Filters in API Gateway:****

Filters in Spring Cloud Gateway can intercept all incoming requests. In this hands-on, a global filter is used to log the details of every incoming request.

#### ****Service Discovery:****

Allows microservices to find and communicate with each other without hardcoding hostname and port. Eureka handles dynamic service registration and lookup.