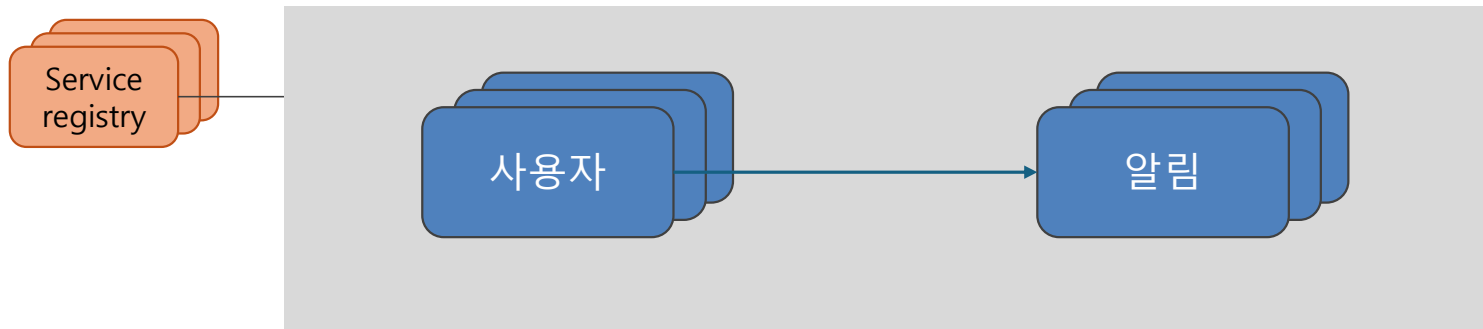


# Service Discovery 적용



Eureka Service Discover 서버를 이용하여  
동적으로 서비스를 찾을 수 있도록 만들어 보자.

# 프로젝트 생성 – eureka project

<https://start.spring.io/> 접속



## Project

☒ Gradle - Groovy

☐ Gradle - Kotlin

☐ Maven

## Language

☒ Java

☐ Kotlin

☐ Groovy

## Spring Boot

☐ 4.0.0 (SNAPSHOT)

☐ 3.5.1 (SNAPSHOT)

☒ 3.5.0

☐ 3.4.7 (SNAPSHOT)

☐ 3.4.6

☐ 3.3.13 (SNAPSHOT)

☐ 3.3.12

## Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 24 ☐ 21 ☒ 17

## Dependencies

ADD DEPENDENCIES... CTRL + B

### Lombok

DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

### Eureka Server

SPRING CLOUD DISCOVERY

spring-cloud-netflix Eureka Server.

# Eureka 프로젝트 설정

---

```
spring:
  application:
    name: backend-eureka
```

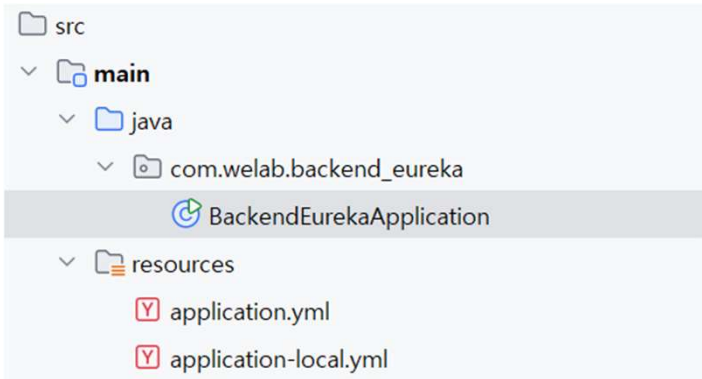
application.yml

```
server:
  port: 8761

# -- Eureka
eureka:
  instance:
    hostname: 127.0.0.1
  client:
    serviceUrl:
      defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
    register-with-eureka: false
    fetch-registry: false
  server:
    eviction-interval-timer-in-ms: 5000 # 인스턴스 제거 간격(5초)
    renewal-percent-threshold: 0.85 # 갱신 임계값 조정
```

application-local.yml

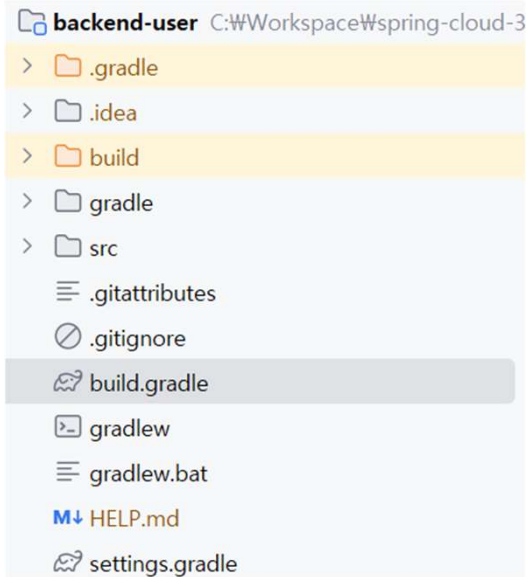
# Eureka 프로젝트 코드 작성



```
@EnableEurekaServer
@SpringBootApplication
public class BackendEurekaApplication {

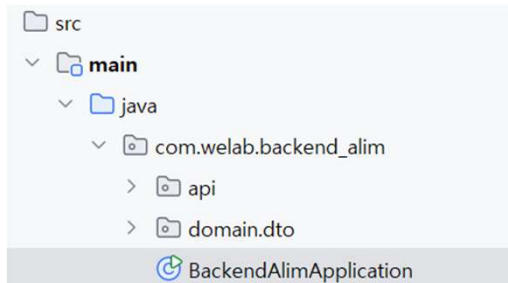
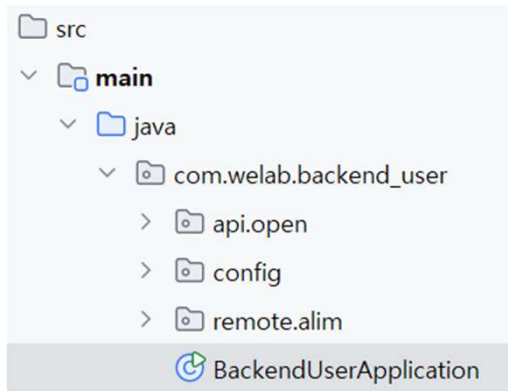
    public static void main(String[] args) {
        SpringApplication.run(BackendEurekaApplication.class, args);
    }
}
```

## User, Alim 프로젝트 의존성 추가



```
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    implementation 'org.springframework.cloud:spring-cloud-starter-loadbalancer'  
    implementation 'org.springframework.cloud:spring-cloud-starter-netflix-eureka-client'  
    compileOnly 'org.projectlombok:lombok'  
    annotationProcessor 'org.projectlombok:lombok'  
  
    testImplementation 'org.springframework.boot:spring-boot-starter-test'  
    testRuntimeOnly 'org.junit.platform:junit-platform-launcher'  
}
```

# User, Alim Application에 어노테이션 추가



```
@EnableDiscoveryClient
@SpringBootApplication
public class BackendUserApplication {

    public static void main(String[] args) {
        SpringApplication.run(BackendUserApplication.class, args);
    }
}
```

## User 프로젝트 설정 변경

Eureka 서버를 이용하도록 설정 변경

```
#spring:
# cloud:
# discovery:
# client:
# simple:
# instances:
#   # 'my-backend-service'에 대한 인스턴스 정의
#   alim-service:
#     - service-id: backend-alim
#       uri: http://localhost:8082
#     - service-id: backend-alim
#       uri: http://localhost:8082

eureka:
  instance:
    prefer-ip-address: true
    lease-renewal-interval-in-seconds: 10 # 하트비트 간격(기본 30초)
    lease-expiration-duration-in-seconds: 30 # 만료 시간(기본 90초)
  client:
    register-with-eureka: true # Discovery Server에 등록
    fetch-registry: true # Discovery Server에서 정보 가져 오
    serviceUrl:
      defaultZone: http://localhost:8761/eureka/ # Discovery Server 위치 지정
```

application-local.yml

# Alim 프로젝트 설정 변경

---

```
server:  
  port: 8082  
  
eureka:  
  instance:  
    prefer-ip-address: true  
    lease-renewal-interval-in-seconds: 10  
    lease-expiration-duration-in-seconds: 30  
  client:  
    register-with-eureka: true  
    fetch-registry: true  
    serviceUrl:  
      defaultZone: http://localhost:8761/eureka/
```


Eureka 서버를 이용하도록 설정 변경



# 서비스 레지스트리 상태 확인

← → ↻ localhost:8761

☰ Miro [LG CNS AM Inspir...

 **spring** Eureka

HOME LAST 1000 SINCE STARTUP

## System Status

Environment	test	Current time	2025-05-26T21:05:59 +0900
Data center	default	Uptime	00:03
		Lease expiration enabled	true
		Renews threshold	5
		Renews (last min)	12

## DS Replicas

### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
BACKEND-ALIM	n/a (1)	(1)	UP (1) - <a href="#">DESKTOP-A0QAITO:backend-alim:8082</a>
BACKEND-USER	n/a (1)	(1)	UP (1) - <a href="#">DESKTOP-A0QAITO:backend-user:8081</a>

## General Info