

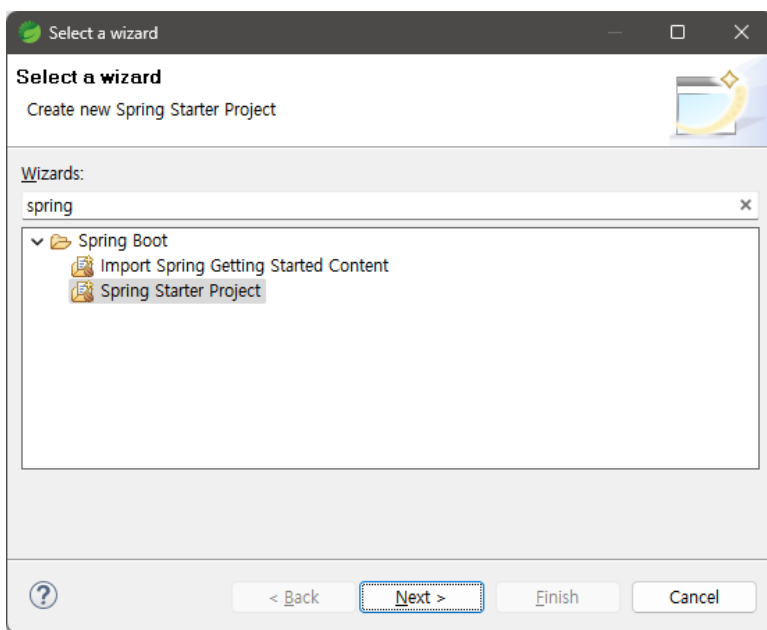


# 05\_Spring Starter를 이용한 프로젝트 생성

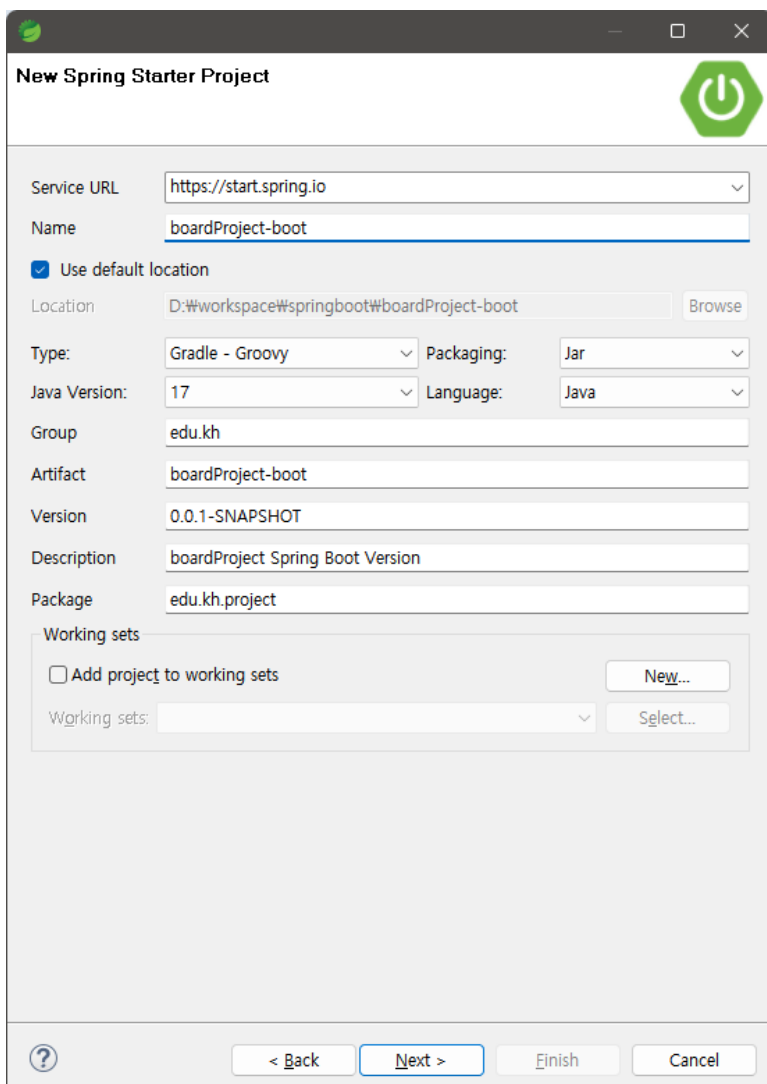
1. STS 마법사(wizard)를 이용해서 Spring Boot 프로젝트 생성
2. 생성할 프로젝트의 기본 설정
3. 프로젝트에서 사용할 모듈, 라이브러리 의존성 추가
4. application.properties 내용 추가
5. 프로젝트 실행 테스트

## 1. STS 마법사(wizard)를 이용해서 Spring Boot 프로젝트 생성

- New → Spring Boot → Spring Starter Project 선택

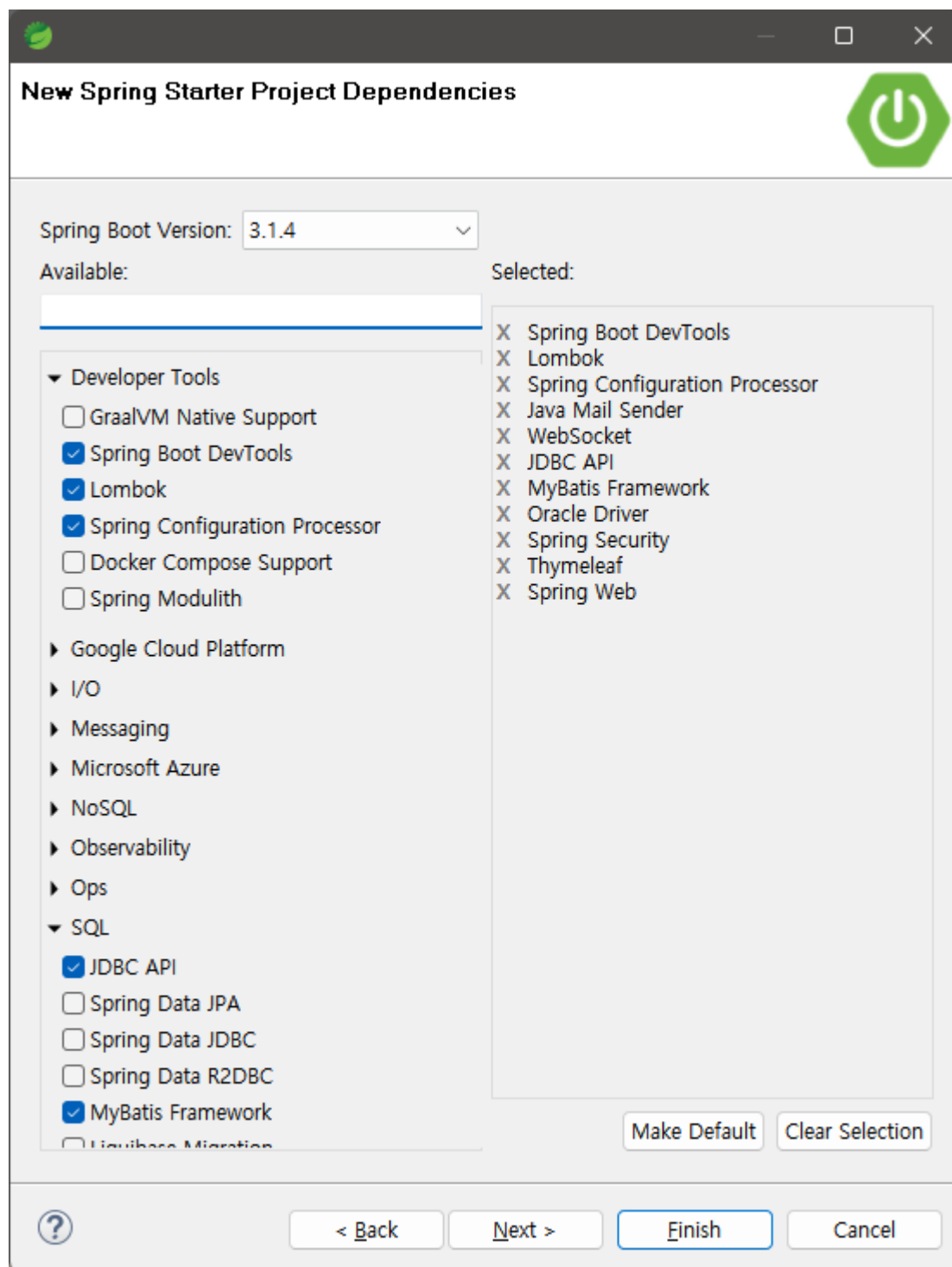


## 2. 생성할 프로젝트의 기본 설정



- 각 항목별 설명
  - Service URL : 데모 프로젝트를 다운 받을 서비스 경로  
(Spring 공식 페이지의 initializr를 통해 설정된 내용대로 데모 프로젝트를 생성해서 받아옴)
  - Name : 프로젝트 이름
  - Type : 빌드 도구(Maven 또는 Gradle)
  - Packaging : 배포 시 압축 형태
  - Java Version : 프로젝트에서 사용할 자바 버전
  - Language : 프로젝트에서 사용할 언어 종류
  - Group : 사용할 그룹(배포시 도메인, 보통 회사명을 작성, 대분류의 개념)
  - Artifact : 만들어진 산출물 이름(프로젝트명, 소분류의 개념)
  - Version : 프로젝트 버전
  - Description : 프로젝트 설명
  - package : 프로젝트 기본 패키지 3레벨 (Group + Artifact로 작성)

### 3. 프로젝트에서 사용할 모듈, 라이브러리 의존성 추가



선택 완료 후 Finish → 다운로드가 진행됨

- Spring Boot Version : 3.2.3(버전은 SNAPSHOT을 제외한 최신 버전 선택)
- Selected
  - Spring Boot DevTools : Spring Boot 개발성 향상을 위한 도구 모음  
(클래스 로딩 문제 진단, 속성 기본값, 자동 재시작, 라이브 리로드, 전역 설정, 원격 애플리케이션)  
→ 브라우저에 표시되는 내용의 코드가 변하면 자동으로 애플리케이션을 재시작하여  
브라우저 내용을 업데이트
  - Lombok : DTO(VO) 자동 완성
  - Spring Configuration Processor : .properties 또는 .yml 파일 자동 완성 지원
  - Java Mail Sender : Java로 메일 발송 코드 지원
  - WebSocket : 전이중 통신 관련 코드 지원(ex : 채팅)
  - JDBC API : Spring에서 JDBC를 사용 가능하게 하는 라이브러리 모음
  - MyBatis Framework : Spring에서 MyBatis를 이용하기 위한 라이브러리 모음
  - Oracle Driver : Oracle DB 연결에 필요한 드라이버 클래스가 포함된 라이브러리
  - Spring Security : Spring 보안 관련 라이브러리 모음(암호화)
  - Thymeleaf : View Template Engine으로 클라이언트 응답 화면을 만들 때 사용하는 템플릿(Thymeleaf 사용을 공식적으로 권장)
  - Spring Web : Spring Web Application에 개발에 필요한 라이브러리 모음

## 4. application.properties 내용 추가

```
#서버포트설정
server.port = 80

#로깅 레벨 지정
logging.level.edu.kh.project=debug

#URL에 jsessionid를 붙이지 않게하는 속성
server.servlet.session.tracking-modes=cookie

#Spring DevTools(자동 리로드, 서버 재시작) 설정
spring.devtools.livereload.enabled=true
spring.devtools.restart.enabled=true
spring.thymeleaf.cache=false

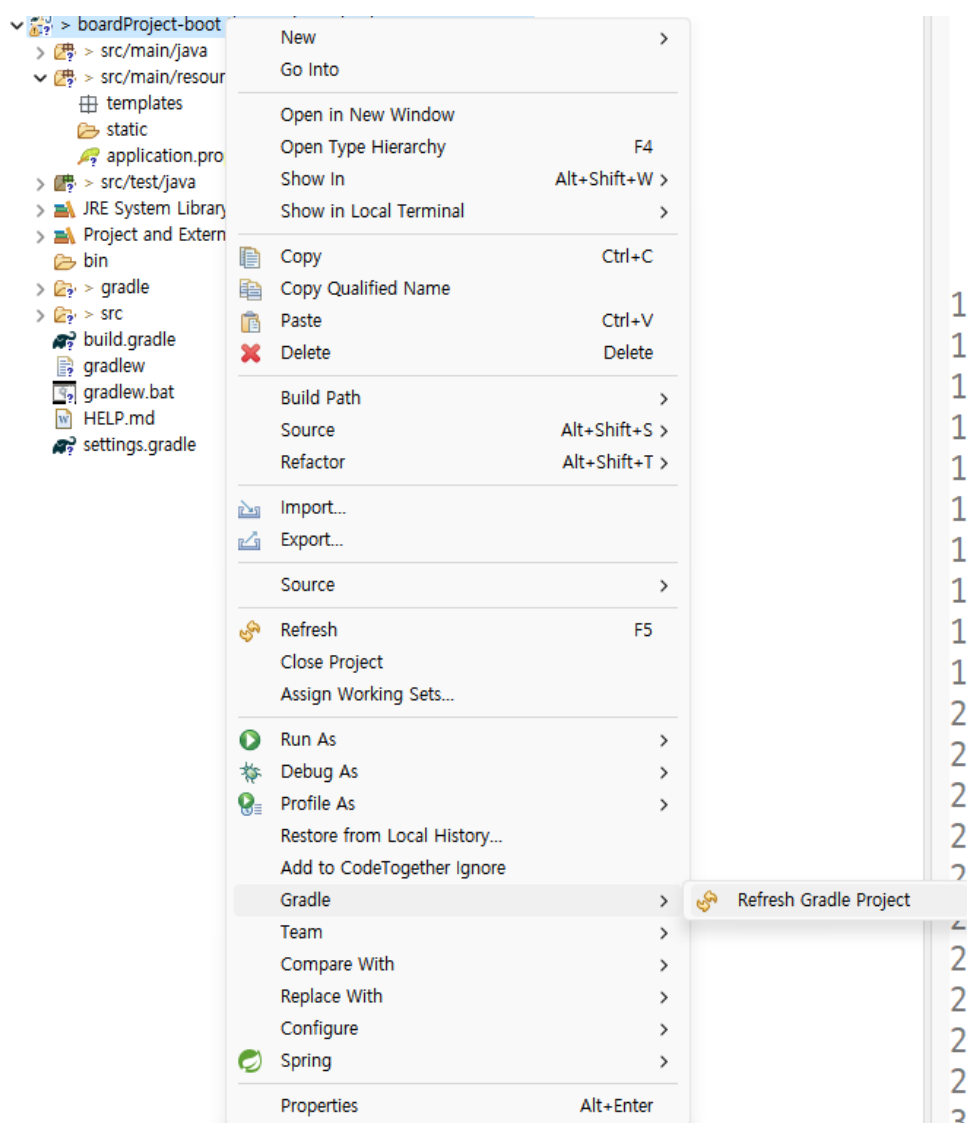
#오라클 DB 연결 정보(해당 파일에 작성하면 DB 연결이 가능하지만 해킹 위험으로 인해 다른 파일에 작성 예정)
#spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
#spring.datasource.url=jdbc:oracle:thin:@IP주소:포트번호/DB이름
```

```
#spring.datasource.username=계정명
#spring.datasource.password=비밀번호
```

## 5. 프로젝트 실행 테스트

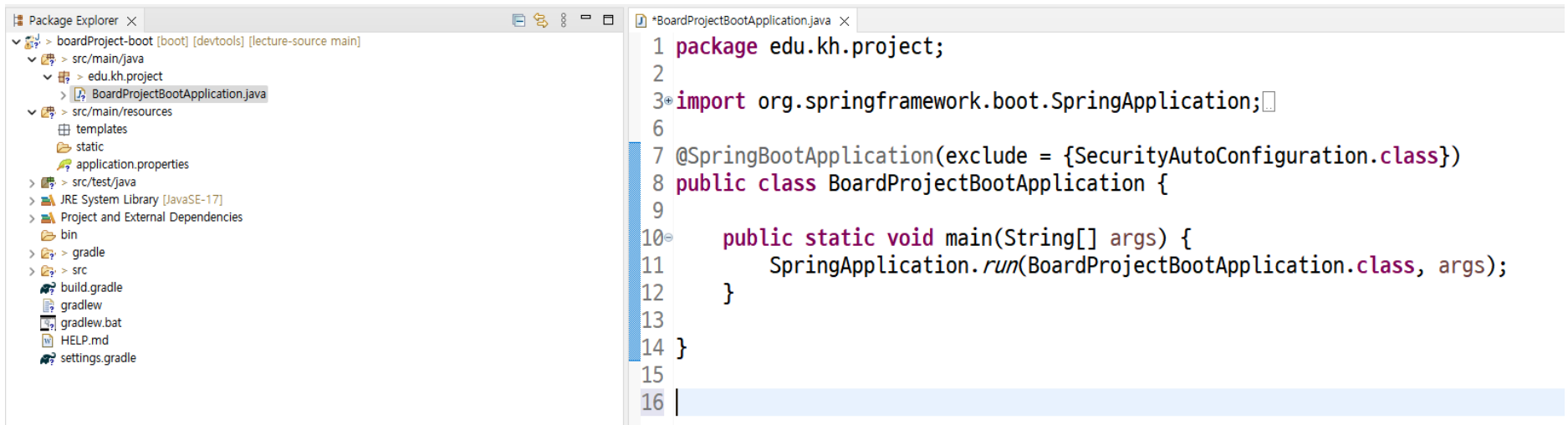
- build.gradle 파일에서 JDBC 관련 dependency 주석 처리 → Refresh Gradle Project 수행

```
24dependencies {
25    //implementation 'org.springframework.boot:spring-boot-starter-jdbc'
26    implementation 'org.springframework.boot:spring-boot-starter-mail'
27    implementation 'org.springframework.boot:spring-boot-starter-security'
28    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
29    implementation 'org.springframework.boot:spring-boot-starter-web'
30    implementation 'org.springframework.boot:spring-boot-starter-websocket'
31    //implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:3.0.2'
32    implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity6'
33    compileOnly 'org.projectlombok:lombok'
34    developmentOnly 'org.springframework.boot:spring-boot-devtools'
35    //runtimeOnly 'com.oracle.database.jdbc:ojdbc8'
36    annotationProcessor 'org.springframework.boot:spring-boot-configuration-processor'
37    annotationProcessor 'org.projectlombok:lombok'
38    testImplementation 'org.springframework.boot:spring-boot-starter-test'
39    testImplementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter-test:3.0.2'
40    testImplementation 'org.springframework.security:spring-security-test'
41}
```

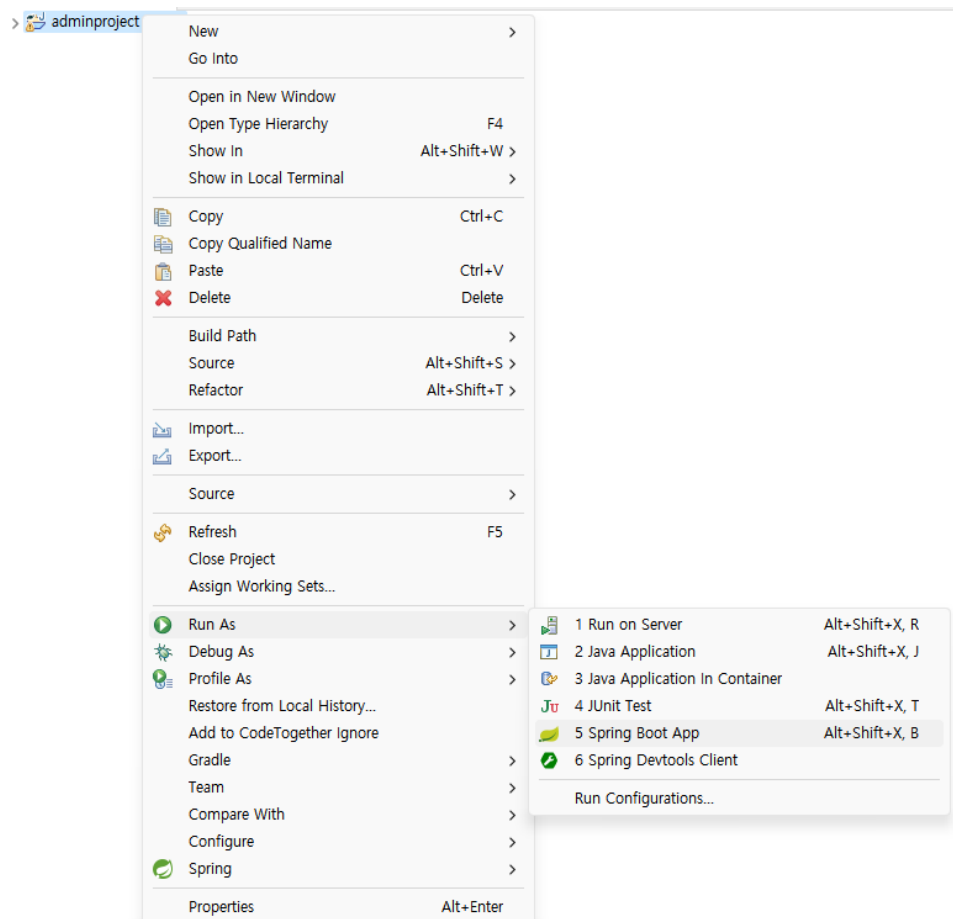


- xxxApplication.java 파일 어노테이션 수정

```
@SpringBootApplication(exclude = {SecurityAutoConfiguration.class}) // import 필요
```

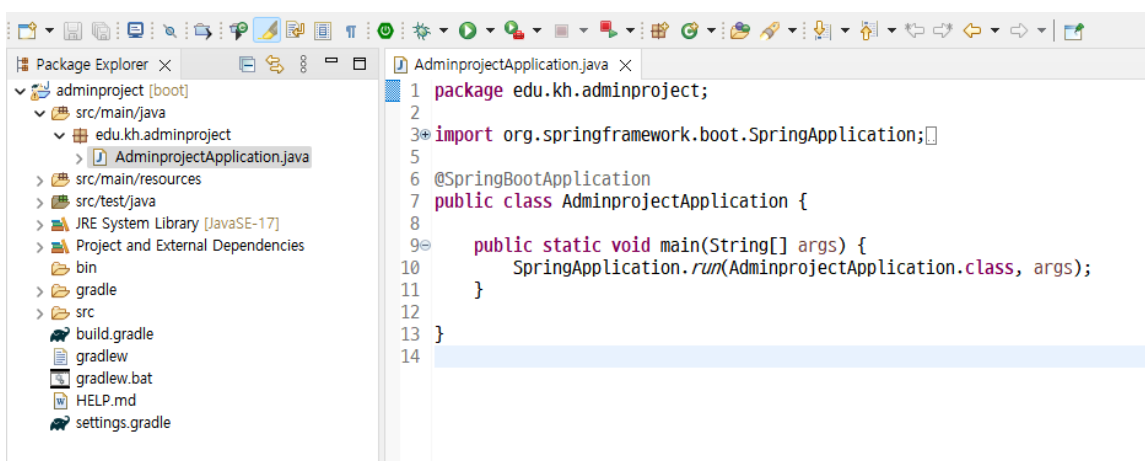


- 프로젝트 우클릭 → Run As → Spring Boot App 클릭

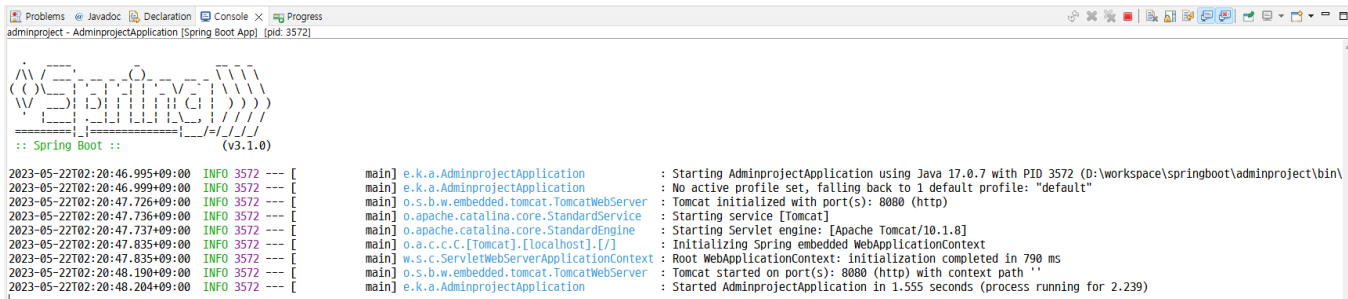


또는

- 자동 생성된 xxxApplication.java 파일에서 Run



- console 실행 로그



```
2023-05-22T02:20:46.995+09:00 INFO 3572 --- [main] e.k.a.AdminprojectApplication : Starting AdminprojectApplication using Java 17.0.7 with PID 3572 (D:\workspace\springboot\adminproject\bin\
2023-05-22T02:20:46.999+09:00 INFO 3572 --- [main] e.k.a.AdminprojectApplication : No active profile set, falling back to 1 default profile: "default"
2023-05-22T02:20:47.726+09:00 INFO 3572 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-05-22T02:20:47.736+09:00 INFO 3572 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-22T02:20:47.737+09:00 INFO 3572 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-05-22T02:20:47.835+09:00 INFO 3572 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-05-22T02:20:47.835+09:00 INFO 3572 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: Initialization completed in 790 ms
2023-05-22T02:20:48.190+09:00 INFO 3572 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-05-22T02:20:48.204+09:00 INFO 3572 --- [main] e.k.a.AdminprojectApplication : Started AdminprojectApplication in 1.555 seconds (process running for 2.239)
```

- <http://localhost/> 접속해서 에러 페이지가 뜨면 성공



# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sun May 28 21:52:44 KST 2023  
There was an unexpected error (type=Not Found, status=404).  
No message available