

SPRING-BOOT 와 JPA를 활용한 쇼핑몰 구현

Version

JDK : 11

spring-boot : 2.3.7 RELEASE

build tool : Gradle

IDE : intellij Ultimate

DBMS : H2 1.4.200

Test : JUnit4

JDK 11 설치

다운로드 : <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

Intellij 설치

다운로드 : <https://www.jetbrains.com/ko-kr/idea/download/>

프로젝트 생성하기

다운로드 : <https://start.spring.io/>

다음과 같이 선택한다. Group과 Artifact ID는 자유롭게 기술하되 GroupId는 모두 소문자, 패키지 경로로 정한다.

Project

☐ Maven Project ☒ Gradle Project

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 2.5.0 (SNAPSHOT) ☐ 2.4.2 (SNAPSHOT) ☐ 2.4.1 ☐ 2.3.8 (SNAPSHOT)

☒ 2.3.7

Project Metadata

Group com.practice

Artifact mymall

Name mymall

Description Demo project for Spring Boot

Package name com.practice.mymall

Packaging ☐ Jar ☒ War

Java ☐ 15 ☒ 11 ☐ 8

Dependencies

ADD ...

No dependency selected

그다음 **Dependency**를 **ADD** 한다. 이 프로젝트에 적용할 서드파티 의존성 라이브러리를 넣을 것이다.

web, thymeleaf, jpa, h2, lombok, validation

Dependencies

ADD ...

Validation **I/O**

JSR-303 validation with Hibernate validator.

Spring Web **WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA **SQL**

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database **SQL**

Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

Lombok **DEVELOPER TOOLS**

Java annotation library which helps to reduce boilerplate code.

Thymeleaf **TEMPLATE ENGINES**

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

하단 **GENERATE** 을 클릭하여 **zip** 파일을 다운 받고, 다운 받은 **zip** 의 압축 풀기

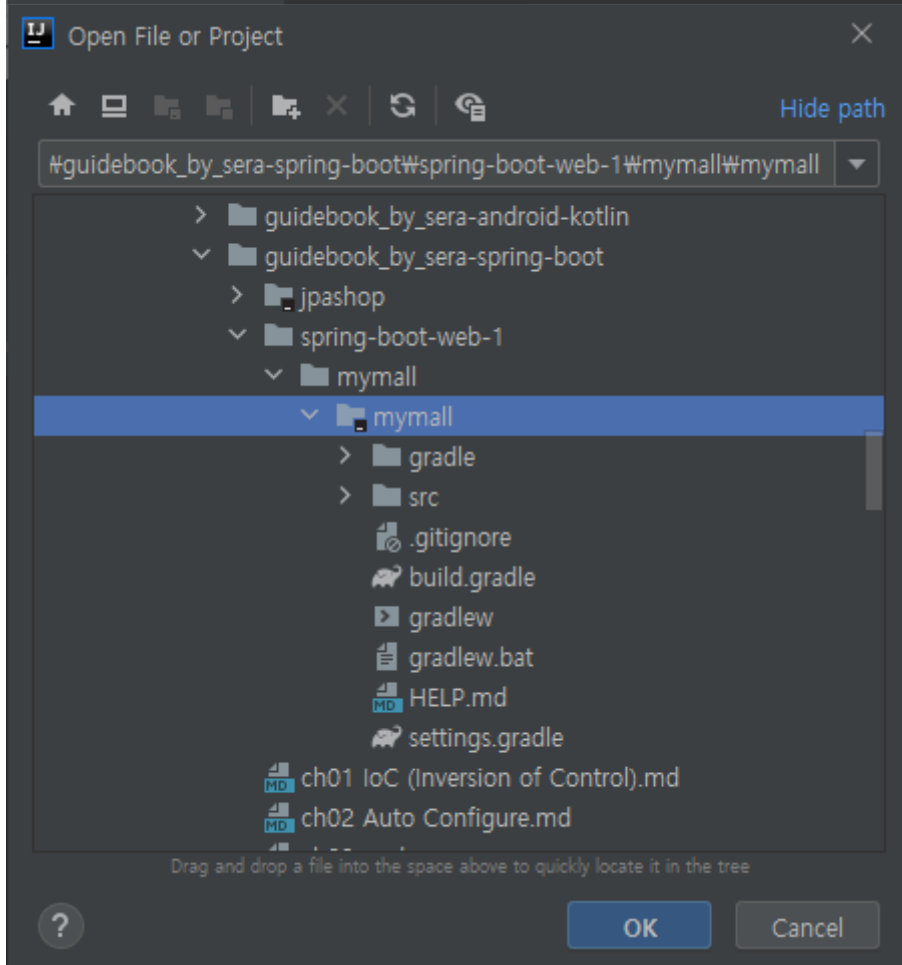
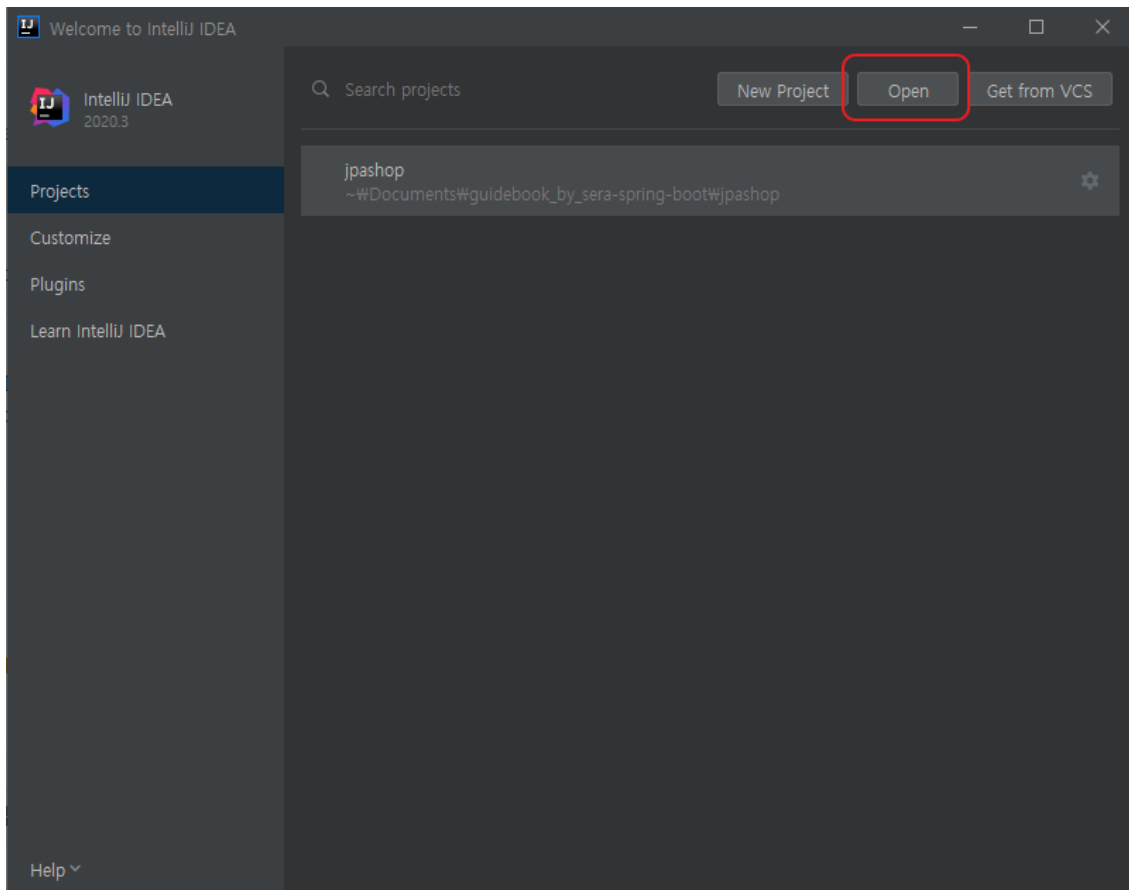


GENERATE

EXPLORE

SHARE...

IntelliJ 에서 Open > Project



프로젝트 설정하기

1. build.gradle 설정

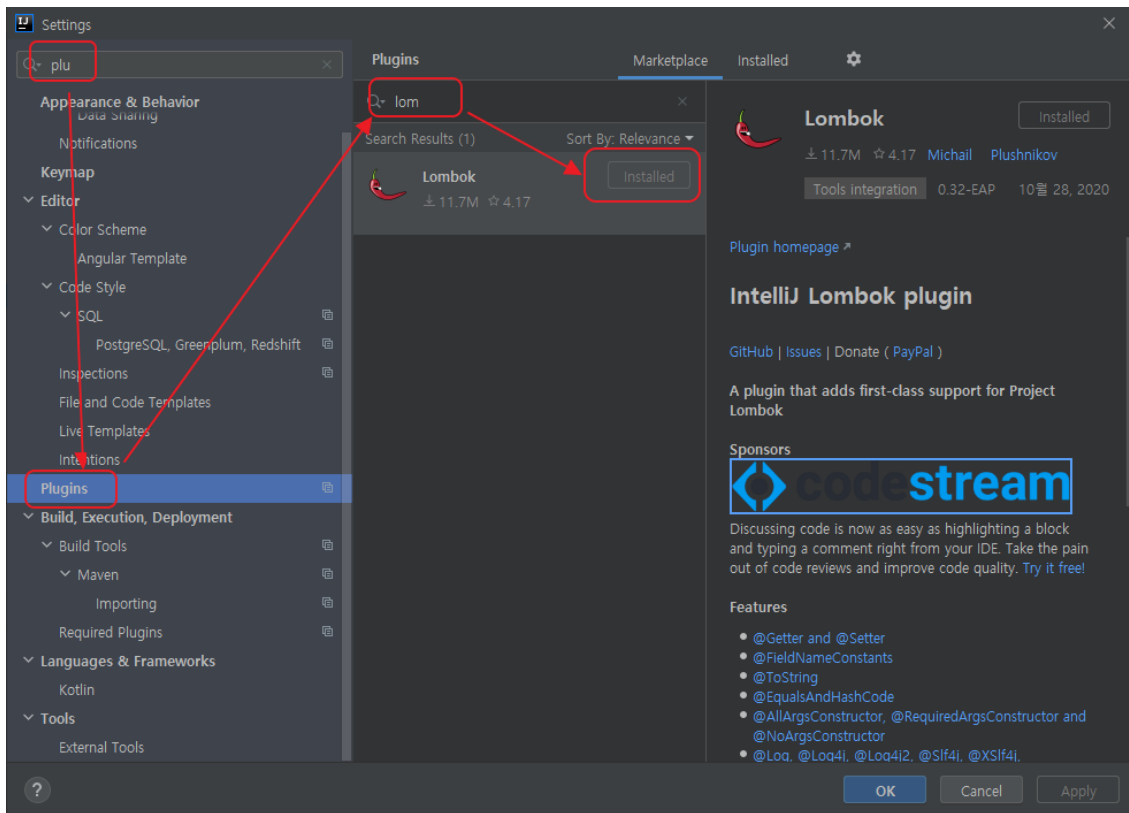
build.gradle에 다음과 같이 작성한다. 기본적으로 이렇게 설정되어있을 것이다.

```
plugins {  
    id 'org.springframework.boot' version '2.3.7.RELEASE'  
    id 'io.spring.dependency-management' version '1.0.10.RELEASE'  
    id 'java'  
    id 'war'  
}  
  
group = 'com.practice'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '11'  
  
configurations {  
    compileOnly {  
        extendsFrom annotationProcessor  
    }  
}  
  
repositories {  
    mavenCentral()  
}  
  
dependencies {  
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'  
    implementation 'org.springframework.boot:spring-boot-starter-validation'  
    implementation 'org.springframework.boot:spring-boot-starter-web'  
    compileOnly 'org.projectlombok:lombok'  
    runtimeOnly 'com.h2database:h2'  
    annotationProcessor 'org.projectlombok:lombok'  
    providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'  
    testImplementation('org.springframework.boot:spring-boot-starter-test') {  
        exclude group: 'org.junit.vintage', module: 'junit-vintage-engine'  
    }  
}  
  
test {  
    useJUnitPlatform()  
}
```

2. Lombok 플러그인 설치

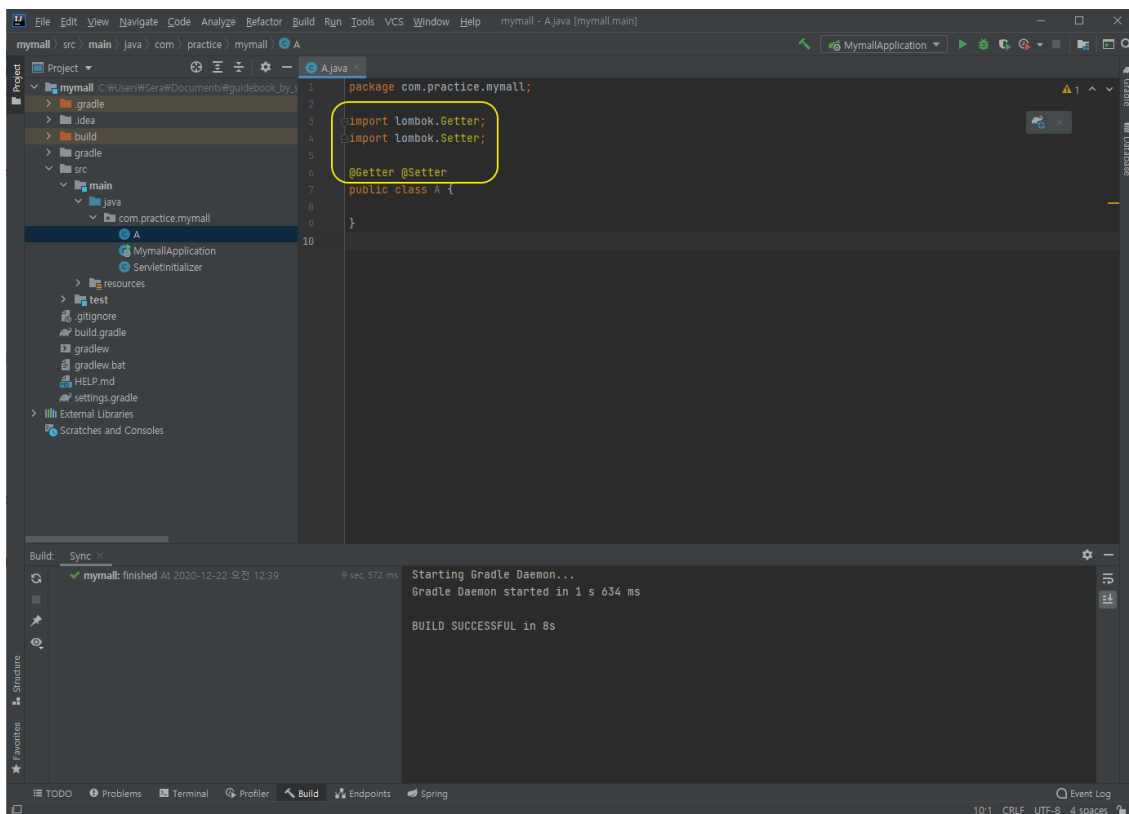
File > Settings > Plugins 검색

Install 누르기



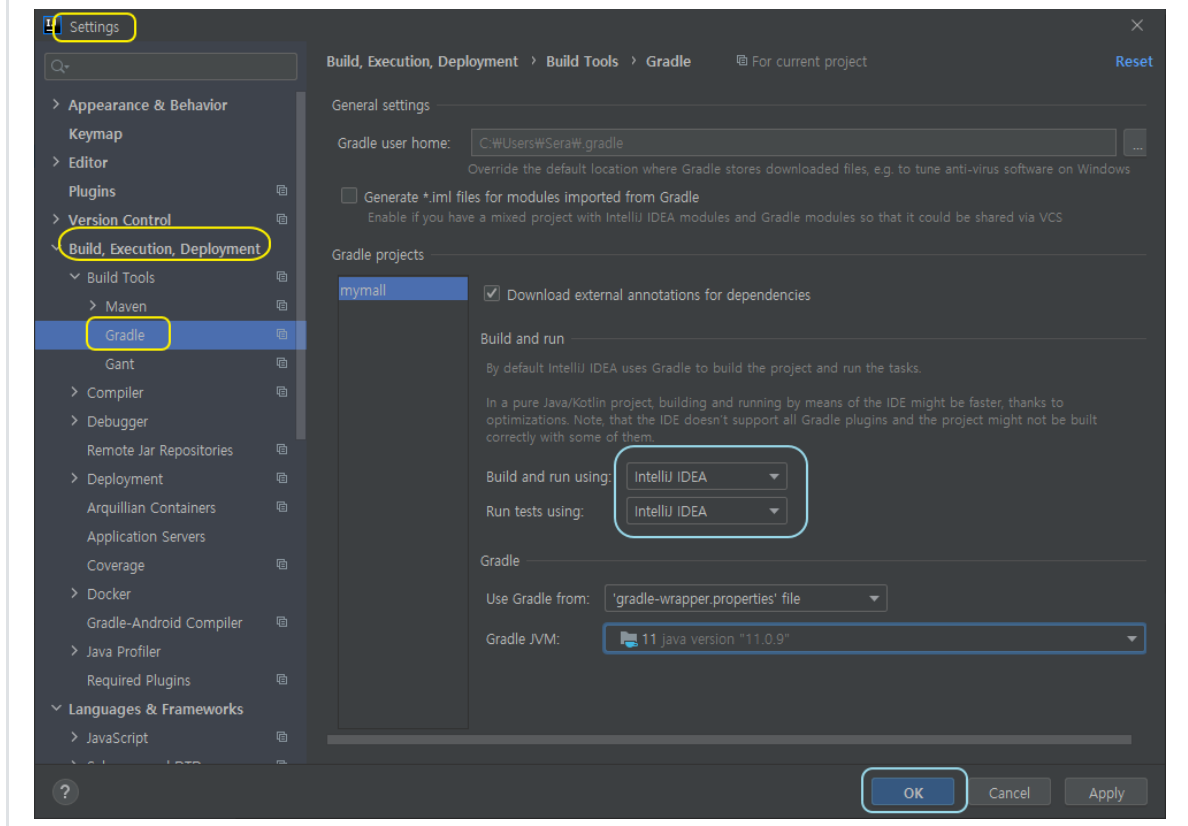
1. Preferences > plugin > lombok 검색 실행 (재시작)
2. Preferences > Annotation Processors 검색 Enable annotation processing 체크 (재시작)
3. 임의의 테스트 클래스를 만들고 @Getter, @Setter 확인

이런식으로 나오면 됨.



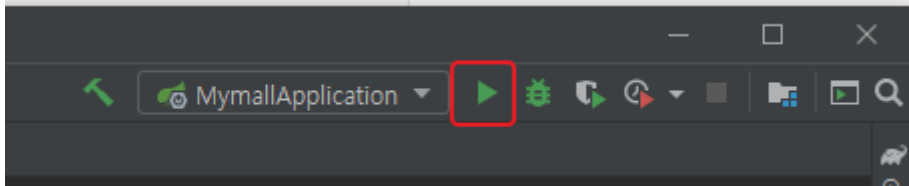
3. IntelliJ Runtime 변경

최신 버전의 IntelliJ 는 기본 VM이 gradle로 되어있다. 이를 JVM으로 변경하자.



4. 실행해보기 (최종)

실행 버튼, 혹은 SHIFT + F10



실행 로그가 출력되고 다음 메시지가 있는지 확인해보자. 여기서 서비스 포트 번호를 확인할 수 있다.

```
[main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
```

<http://127.0.0.1:8080> 으로 접속해보자. `Whitelabel Error` 가 뜬다면 성공!

