

# |. 스프링 프레임워크 개요 및 개발환경 구축

1. 스프링 프레임워크 개요
2. 개발환경 구축
3. Maven 프로젝트
4. 스프링 IoC 컨테이너

## 1.1 프레임워크의 장점

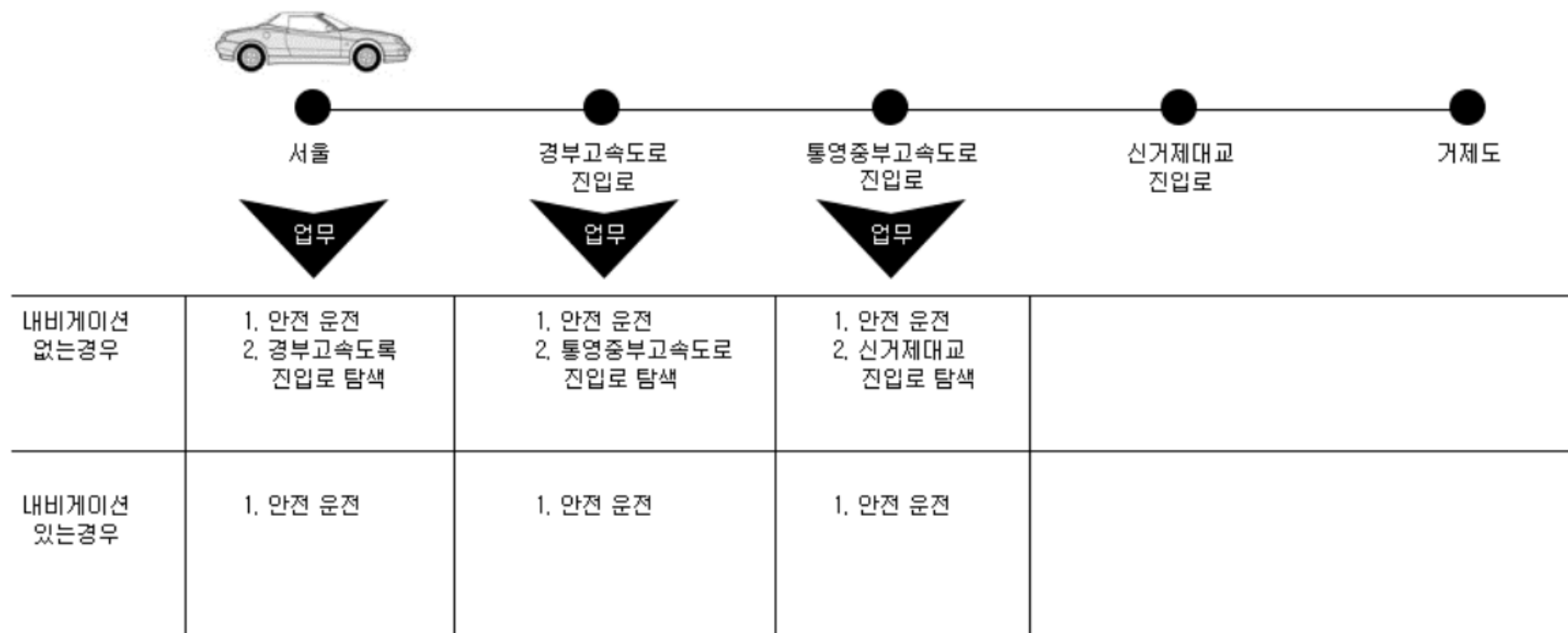
- 빠른구현시간
  - 비기능업무(성능, 보안, 확장성, 안정성등 공통 로직)는 프레임워크가 제공
  - 개발자들은 비즈니스 로직만 구현하면 됨
- 쉬운관리
  - 같은 프레임워크가 적용된 애플리케이션들은 아키텍처가 같으므로 관리가 수월.
  - 유지보수 인력과 시간을 줄일 수 있음
  - 프레임워크를 사용하면 숙련된 개발자와 초급 개발자가 생성한 코드가 비슷해짐.
- 검증된 아키텍처의 재사용과 일관성 유지

## I.2 Spring 프레임워크 주요 특징

- POJO(Plain Old Java Object) 기반 구성
  - Spring 컨테이너에 저장되는 Java객체는 특정한 인터페이스를 구현하거나, 특정 클래스를 상속받지 않아도 된다.
- 의존성 주입(DI)을 통한 객체간의 관계구성
- AOP 지원
  - 트랜잭션 처리 지원
- 편리한 MVC 구조
- 테스트 환경
  - **junit**를 이용하여 쉬운 테스트 환경을 제공한다.
- 전자정부 표준프레임워크의 기반 기술
- 오픈소스
  - Java 엔터프라이즈 개발을 편하게 해주는 오픈소스 경량급 애플리케이션 프레임워크이다.
  - 스프링 프레임워크 개발자는 로드존슨 2002년 발표했으며 유지보수는 Pivotal (VMware 자회사)에서 함.

## 1.3 Spring 프레임워크 모듈

- 스프링 프레임워크는 주요기능으로 DI, AOP, MVC, JDBC 모듈 등을 제공한다.

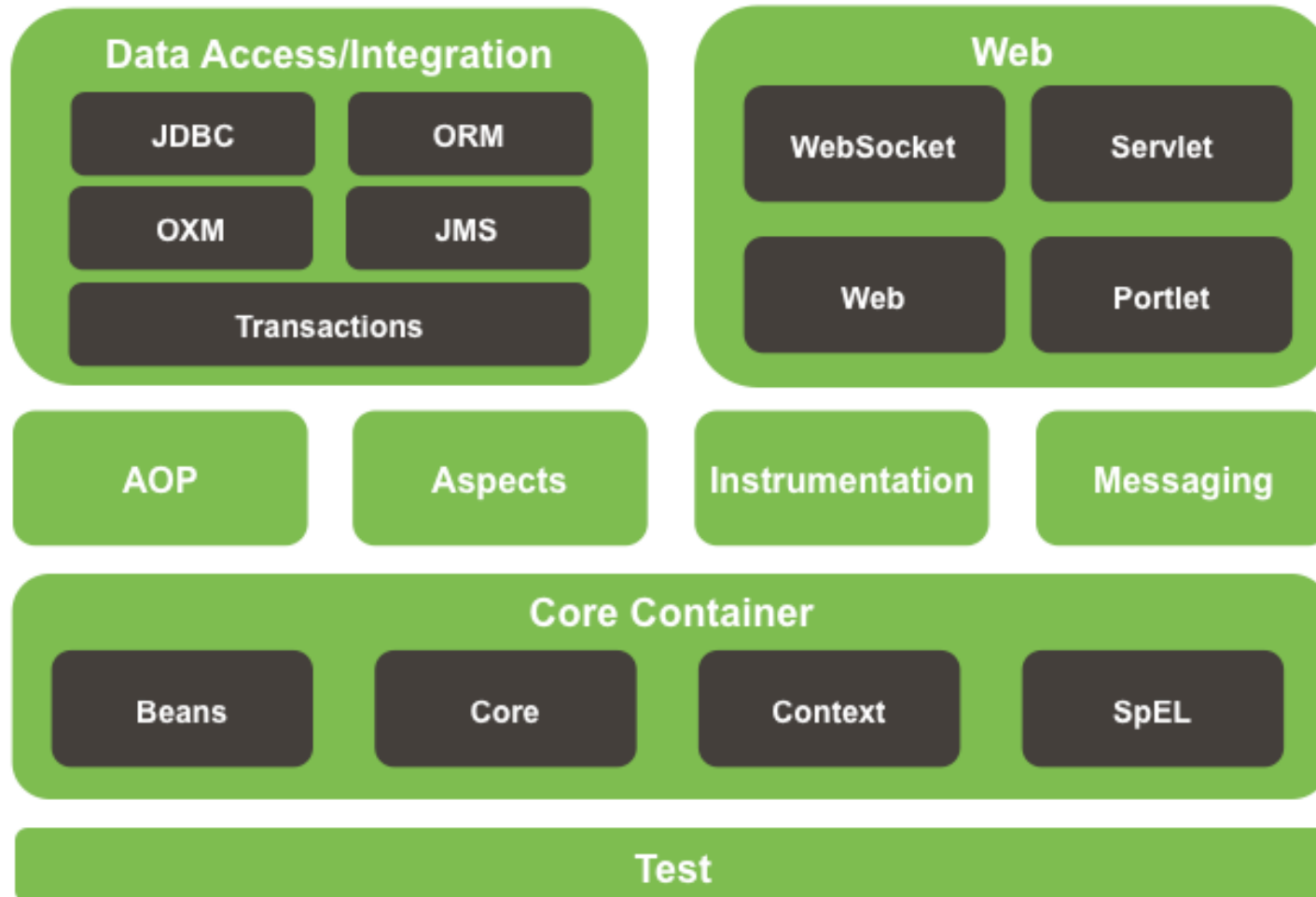


(내비게이션 無 : 내비게이션 有 = 다양한 업무 : 본연의 업무)

## 1.3 Spring 프레임워크 모듈

스프링 모듈	기능
spring-core	스프링의 핵심인 DI(Dependency Injection)와 IoC(Inversion of Control)를 제공 제어의 역전. 인스턴스 생성부터 소멸까지의 인스턴스 생명주기 관리를 개발자가 아닌 컨테이너가 대신 해줌.
spring-aop	Spring은 트랜잭션이나 로깅, 보안과 같이 공통적으로 필요로 하는 모듈들을 실제 핵심 모듈에서 분리해 서 적용할 수 있다
spring-jdbc	데이터베이스를 쉽게(적은 양의 코드) 다룰 수 있는 기능 제공
spring-tx	스프링에서 제공하는 트랜잭션 관련 기능 제공
spring-webmvc	스프링에서 제공하는 컨트롤러(Controller)와 뷰(View)를 이용한 스프링MVC 구현 기능 제공

## 1.3 Spring 프레임워크 모듈



## 2.1 프로그램 설치 순서

- JDK 11 설치
- 환경변수 설정 : JAVA\_HOME, Path
- 이클립스 설치
- 이클립스 플러그인 설치
- Lombok 설정
- Tomcat 9.0 설치

## 2.2 JDK 11 설치

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

Java downloads Tools and resources Java archive

### Java SE subscribers have more choices

Also available for development, personal use, and to run other licensed Oracle products.

Java 8 **Java 11**

### Java SE Development Kit 11.0.16.1

Java SE subscribers will receive JDK 11 updates until at least **September of 2026**.

These downloads can be used for development, personal use, or to run Oracle licensed products. Use for other purposes, including production or commercial use, requires a Java SE subscription or another Oracle license.

JDK 11 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#).

JDK 11.0.16.1 [checksum](#)

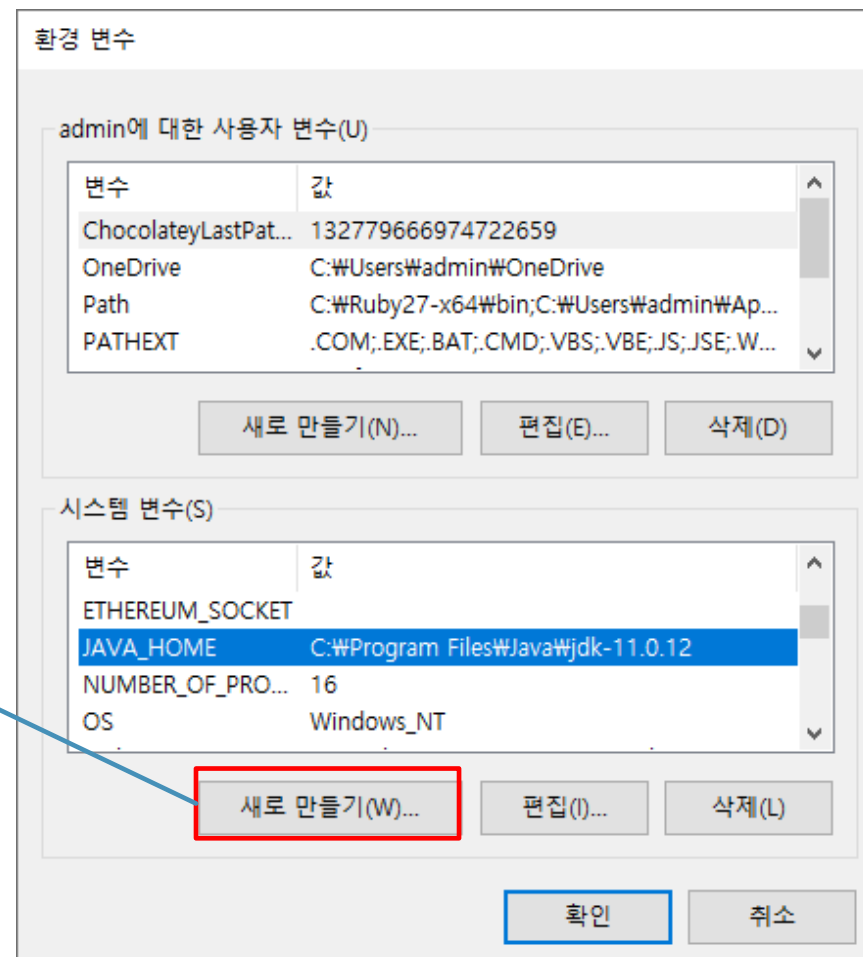
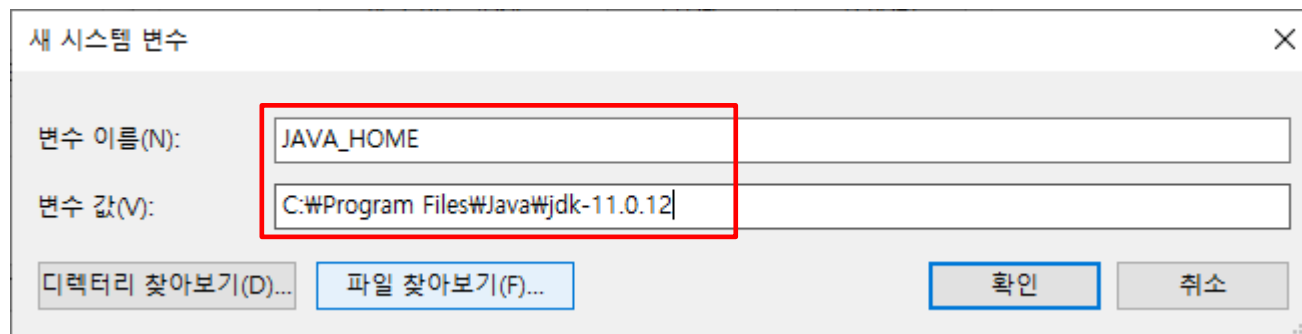
Linux macOS Solaris **Windows**

Product/file description	File size	Download
x64 Installer	140.55 MB	<a href="#">jdk-11.0.16.1_wind</a>
x64 Compressed Archive	158.30 MB	<a href="#">jdk-11.0.16.1_wind</a>



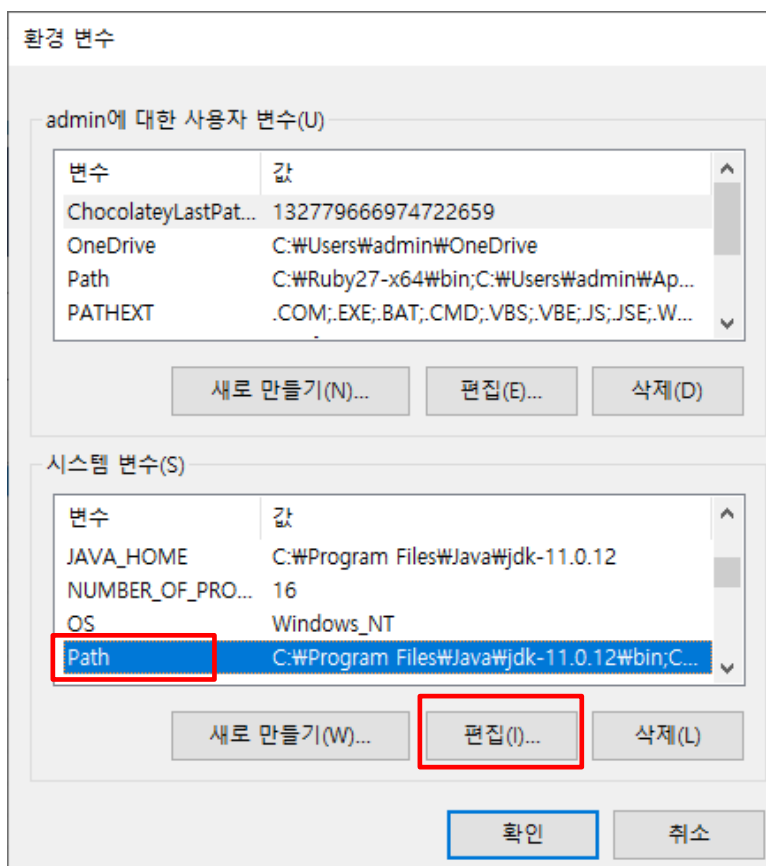
## 2.3 환경변수 설정

- 시스템 환경변수 "JAVA\_HOME" 추가
  - 시스템 환경 변수 편집 실행
  - [고급] 탭에서 환경 변수 클릭
  - [새로 만들기] 클릭
    - 변수이름: JAVA\_HOME
    - 변수 값: C:\Program Files\Java\jdk-11.0.12

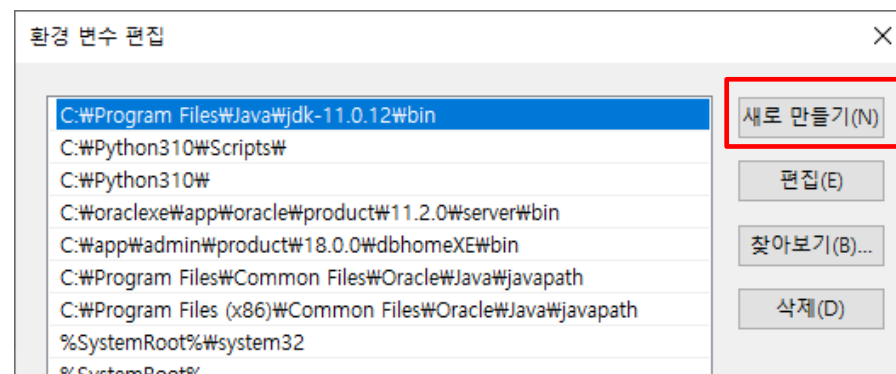


## 2.3 환경변수 설정

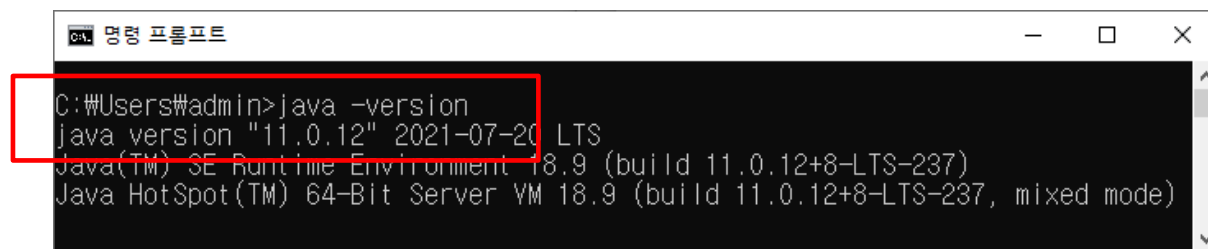
### ■ 시스템 환경변수 "PATH" 에 자바 실행 경로 추가



Path 편집




Path 설정 확인



## 2.4 Eclipse 설치

- 다운로드: <https://www.eclipse.org/downloads/packages/>

최신 버전



**Eclipse IDE for Enterprise Java and Web Developers**

495 MB 308,286 DOWNLOADS

Tools for developers working with Java and Web applications, including a Java IDE, tools for JavaScript, TypeScript, JavaServer Pages and Faces, Yaml, Markdown, Web Services, JPA and Data Tools, Maven and Gradle, Git, and more.

Click [here](#) to file a bug against Eclipse Web Tools Platform.

Click [here](#) to file a bug against Eclipse Platform.

Click [here](#) to file a bug against Maven integration for web projects.

Click [here](#) to report an issue against Eclipse Wild Web Developer (incubating).

Windows  
x86\_64  
macOS x86\_64  
| AArch64  
Linux x86\_64 |  
AArch64


**RELATED LINKS**

- [Compare & Combine Packages](#)
- [New and Noteworthy](#)
- [Install Guide](#)
- [Documentation](#)
- [Updating Eclipse](#)
- [Forums](#)
- [Simultaneous Release](#)

**MORE DOWNLOADS**

- [Other builds](#)
- [Eclipse 2022-06 \(4.24\)](#)
- [Eclipse 2022-03 \(4.23\)](#)
- [Eclipse 2021-12 \(4.22\)](#)
- [Eclipse 2021-09 \(4.21\)](#)
- [Eclipse 2021-06 \(4.20\)](#)
- [Eclipse 2021-03 \(4.19\)](#)
- [Eclipse 2020-12 \(4.18\)](#)

이전 버전



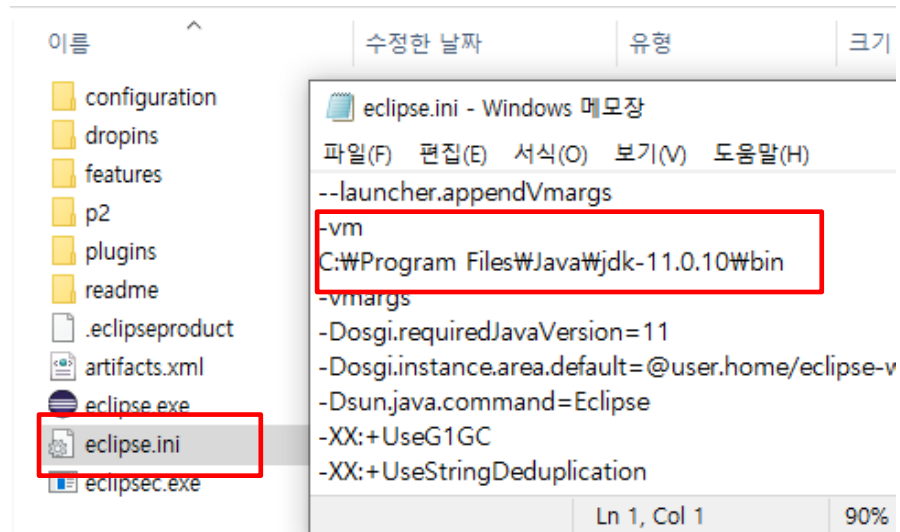
**Eclipse IDE for C/C++ Developers**

340 MB 70,959 DOWNLOADS

Windows  
x86\_64  
macOS x86\_64  
| AArch64  
Linux x86\_64 |

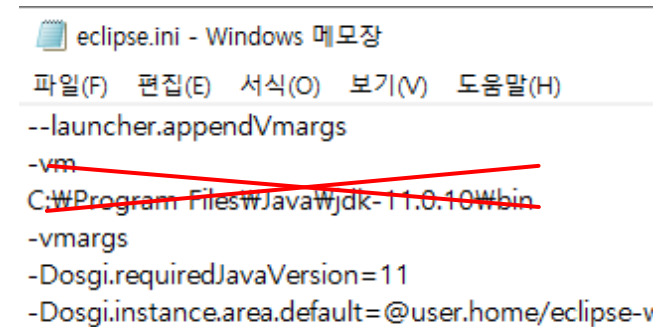
## 2.4 Eclipse 설치

- 압축해제 : c:/dev
- 이클립스 설정파일에 java 실행경로 지정
  - c:/dev/eclipse/eclipse.ini



-vm 옵션을 변경

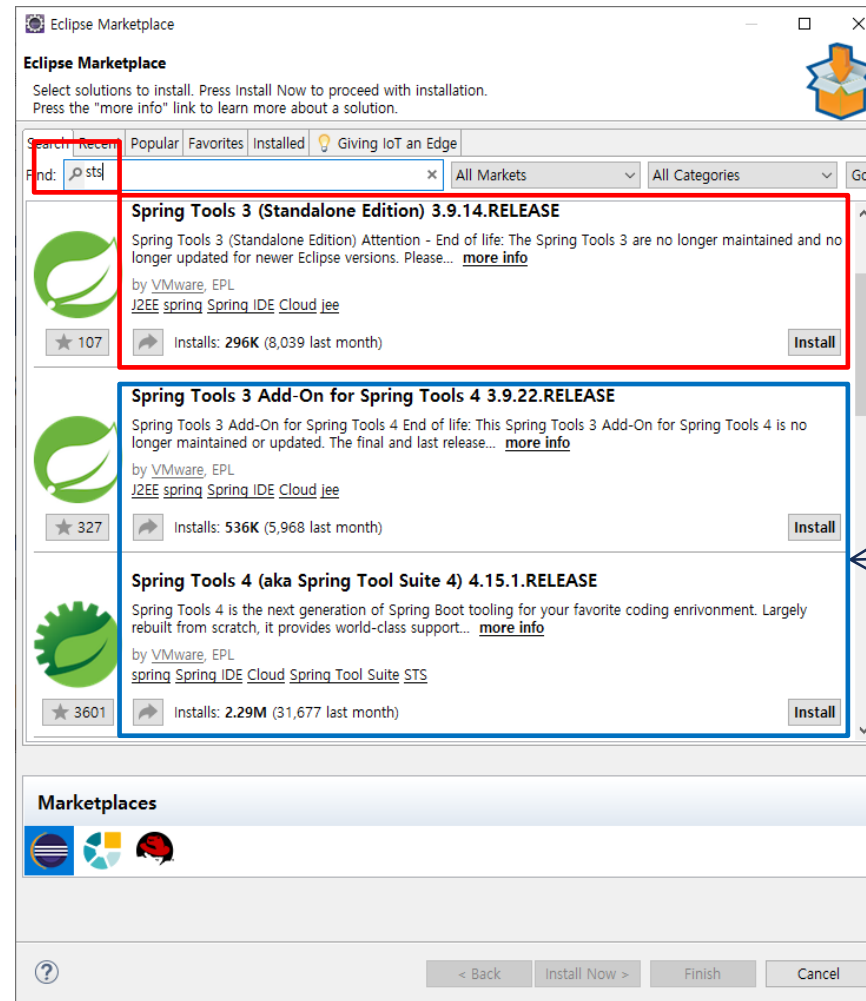
또는



-vm 삭제

## 2.5 이클립스 플러그인 설치

- STS 플러그인 설치
  - Eclipse Marketplace에서 sts 검색
- Spring Tools 3 설치
  - Spring Legacy : xml 기반, AOP 지원
  - eclipse용 스프링 도구
  - 또는 Spring Tools 4버전 설치하고 Spring Tools 3 Add-On 설치

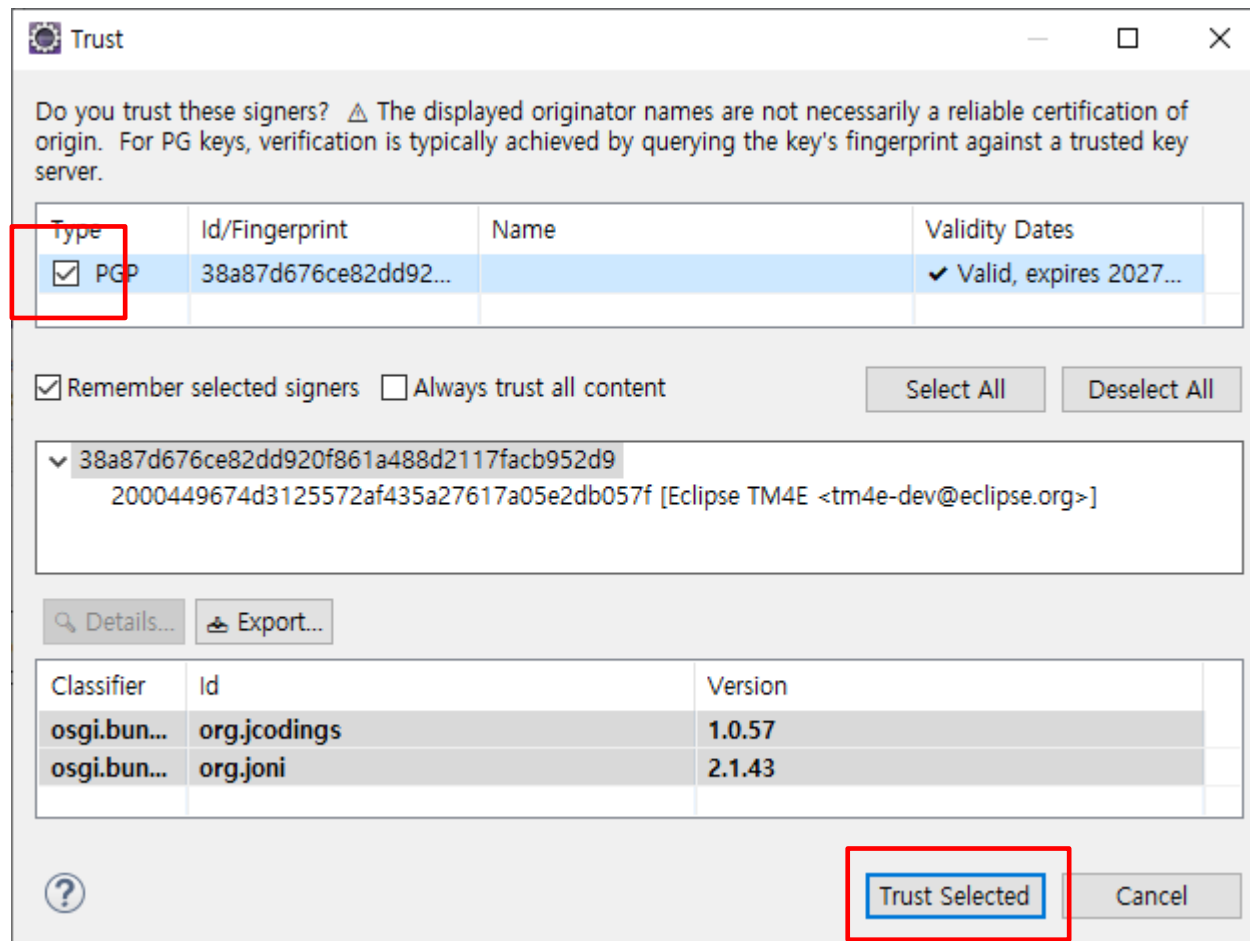


spring Tools 4는 Spring Boot 만 지원함

spring tools 3 Add-On을 설치해야 spring legacy를 사용할 수 있음

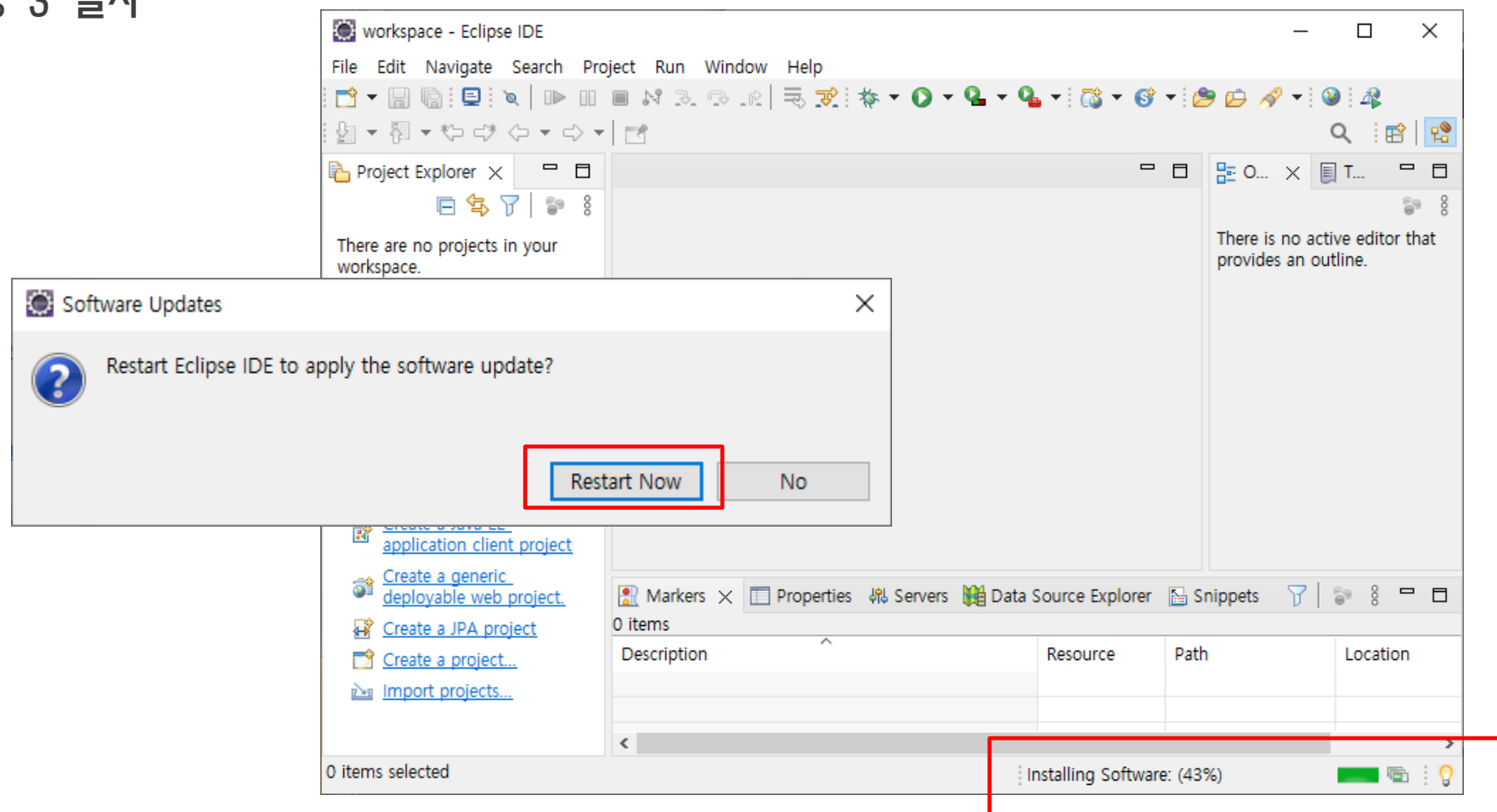
## 2.5 이클립스 플러그인 설치

### ■ Spring Tools 3 설치



## 2.5 이클립스 플러그인 설치

### ■ Spring Tools 3 설치



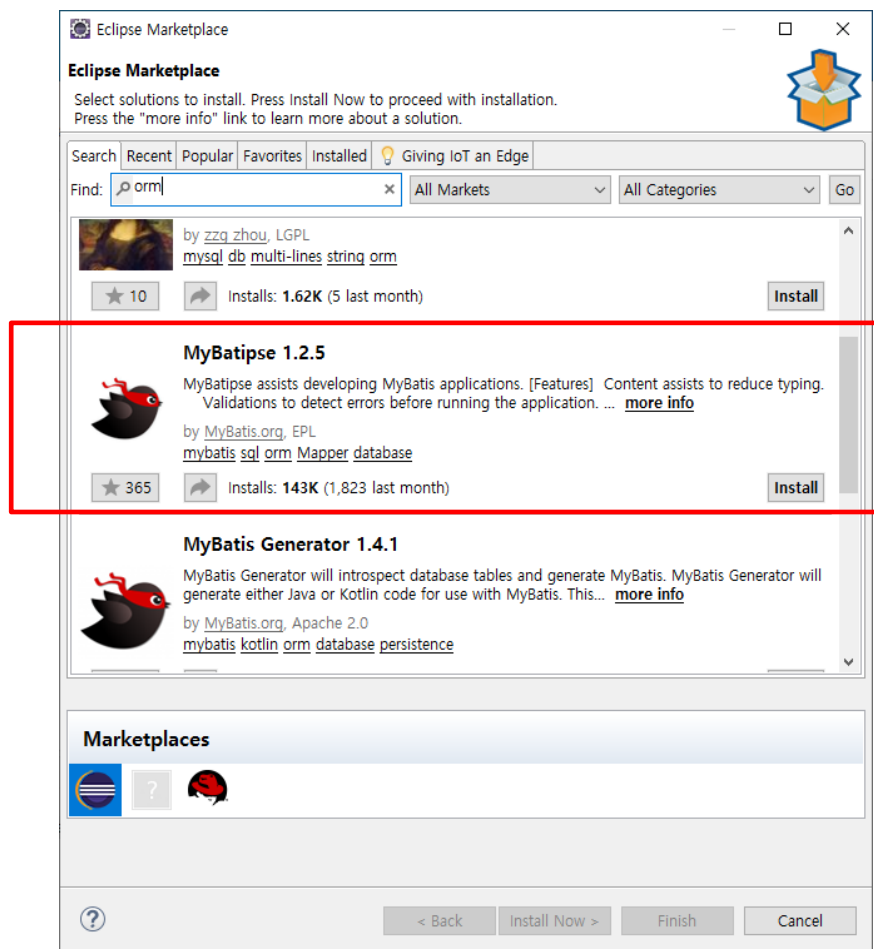
## 2.5 이클립스 플러그인 설치

### ■ ORM 플러그인 설치

- Mybatis와 관련된 복잡한 XML 설정 파일들을 자동으로 만들고 관리

### ■ 설치순서

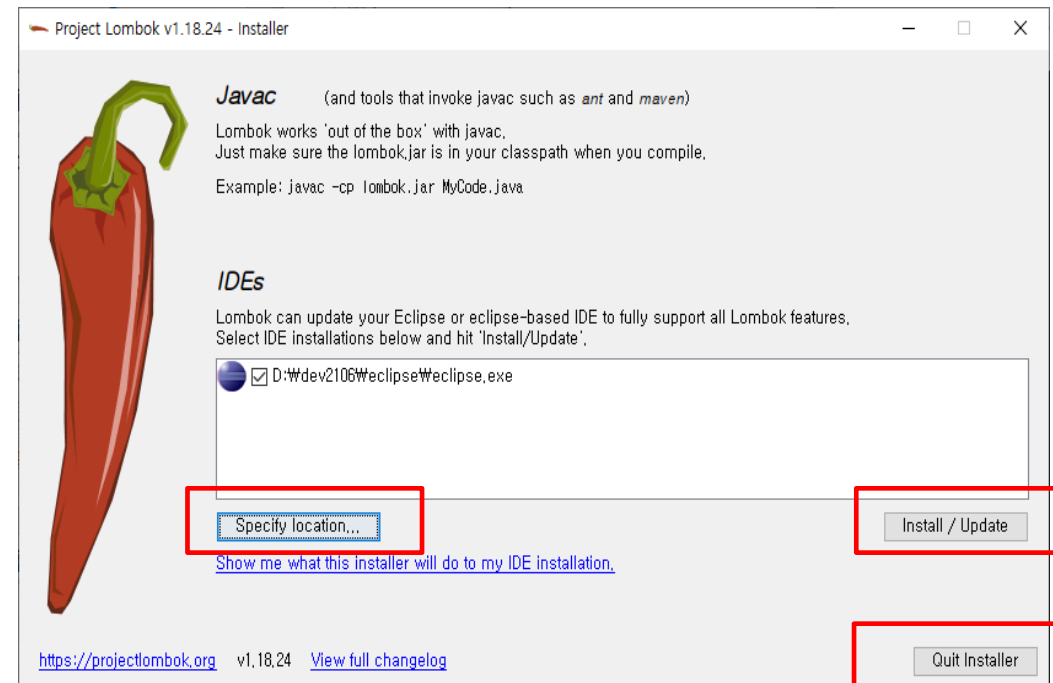
- Marketplace에서 ORM 검색
- MyBatipse Install
- install anyway
- restart





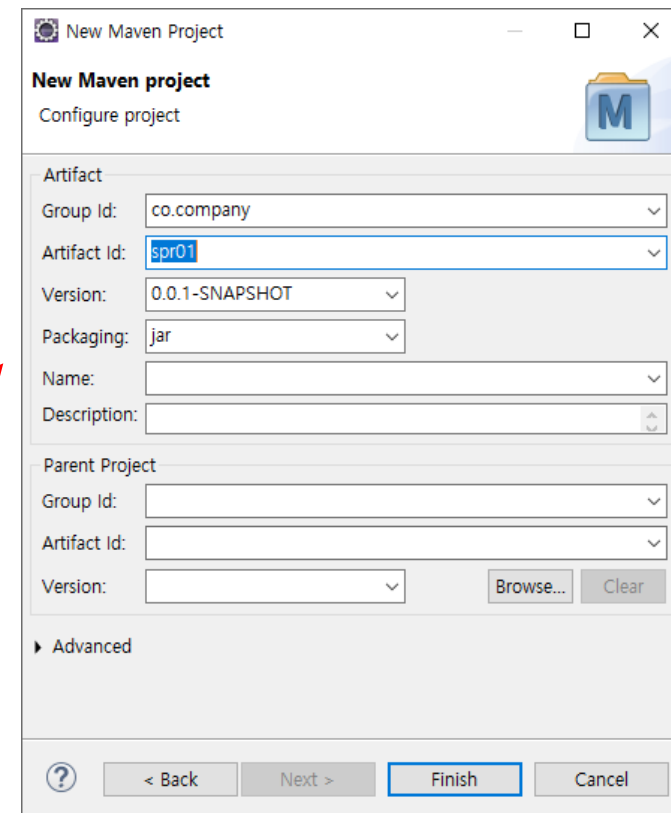
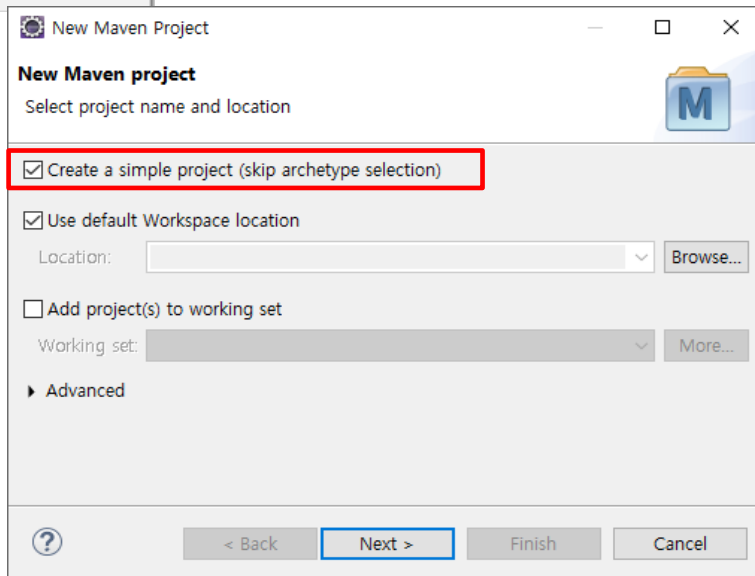
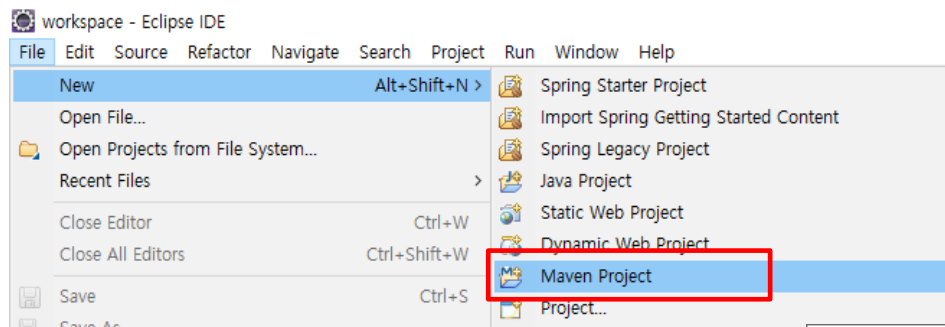
## 2.6 Lombok 설치

- Java 개발 시 자주 사용되는 getter/setter, toString(), 생성자 등을 자동으로 생성
- 설치 순서
  - lombok.jar 실행 : Lombok.jar 파일을 더블클릭하거나 java 명령어를 이용하여 실행.
    - C:\Users\admin\.m2\repository\org\projectlombok\lombok\1.18.24> `java -jar lombok-1.18.24.jar`
  - Sepecify location 클릭
    - 이클립스 실행파일 위치 등록
  - install/update 클릭
  - Quit Installer 클릭
- 메뉴얼 : <https://projectlombok.org/features/>



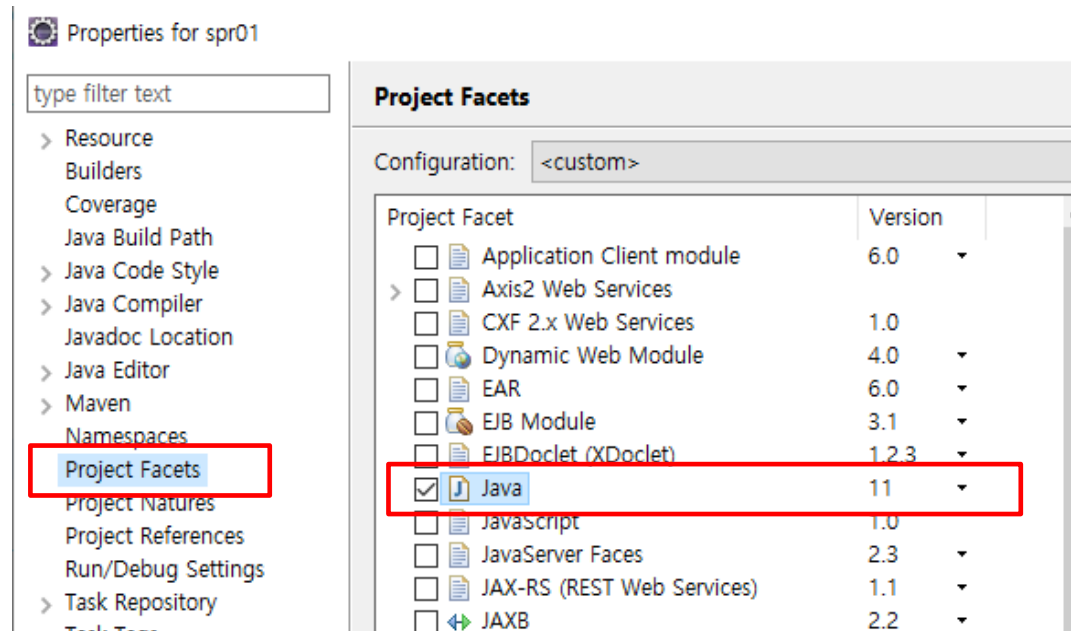
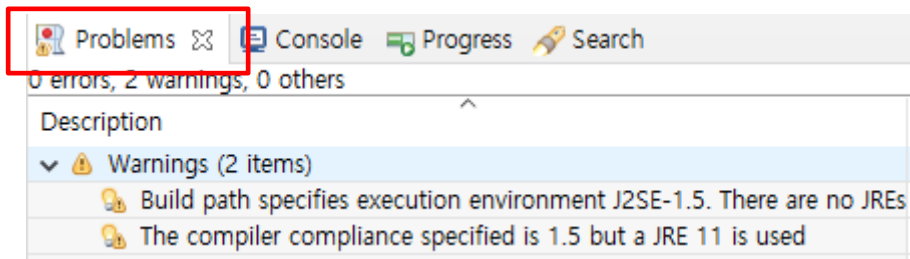
## 3.1 Maven 프로젝트로 스프링 프로젝트 만들기

### ■ 프로젝트 생성



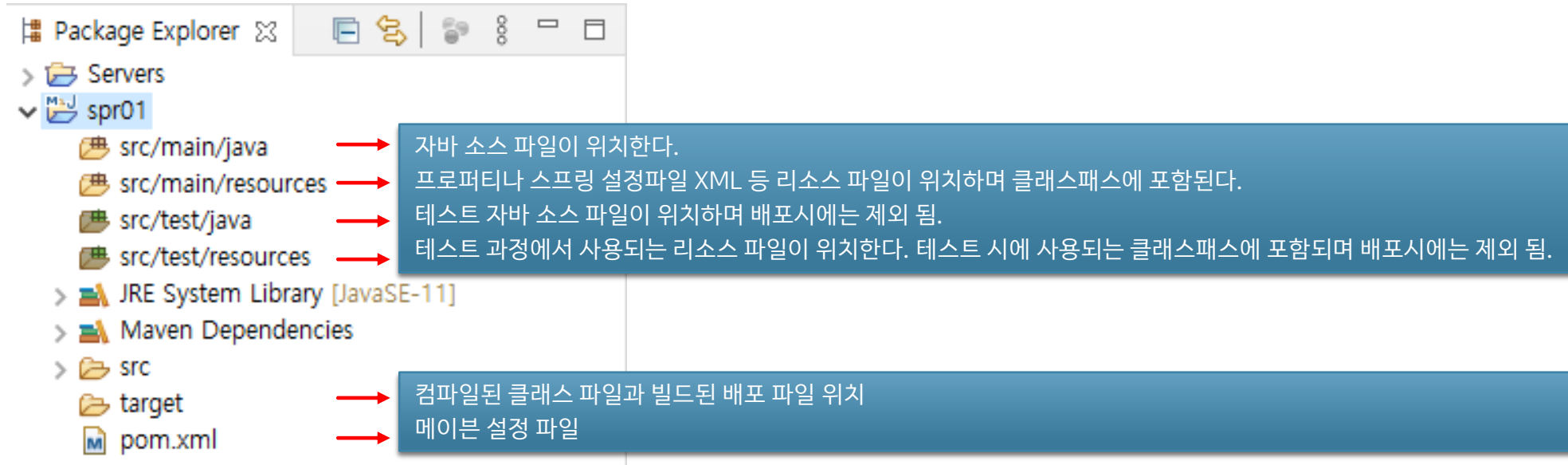
## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

- 자바 버전 변경
  - 자바버전에 대한 경고 확인
  - 프로젝트 Properties 창에서 [Project Facets] 선택
  - Java 버전을 11로 변경하면 경고가 사라짐



## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

### ■ Maven 프로젝트 폴더 구성



## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

- 소스폴더가 보이지 않는다면 소스 폴더 추가
  - 1. 폴더 생성

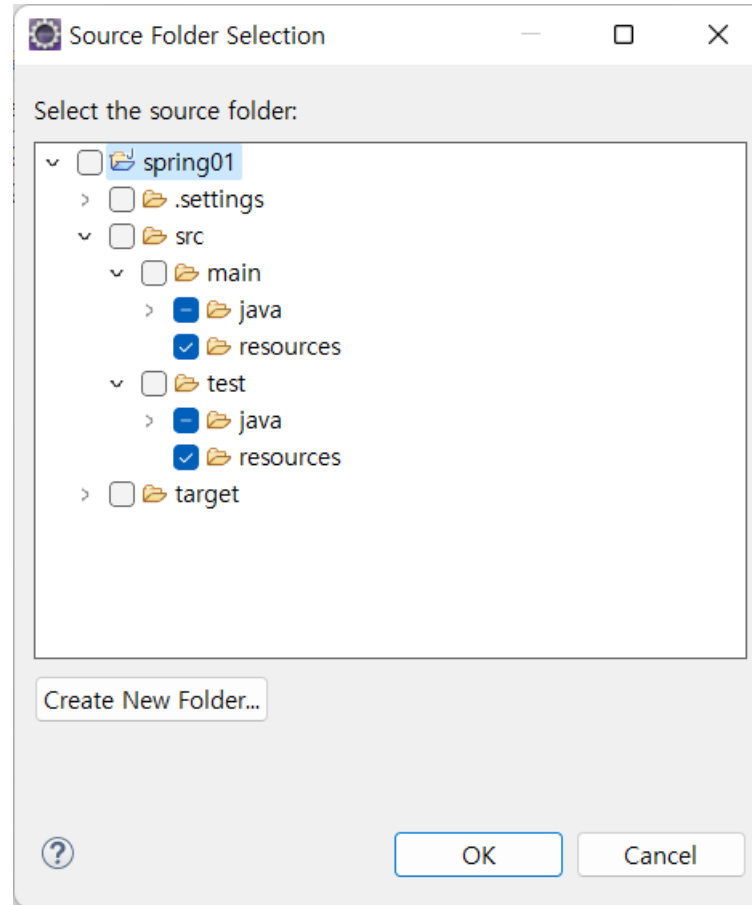
```
C:\dev\workspace\spr01>tree src /f
```

```
C:\DEV\WORKSPACE\SPR01\SRC
```

```
|—main
|   |—java
|   |—resources
|—test
|   |—java
|   |—resources
```

## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

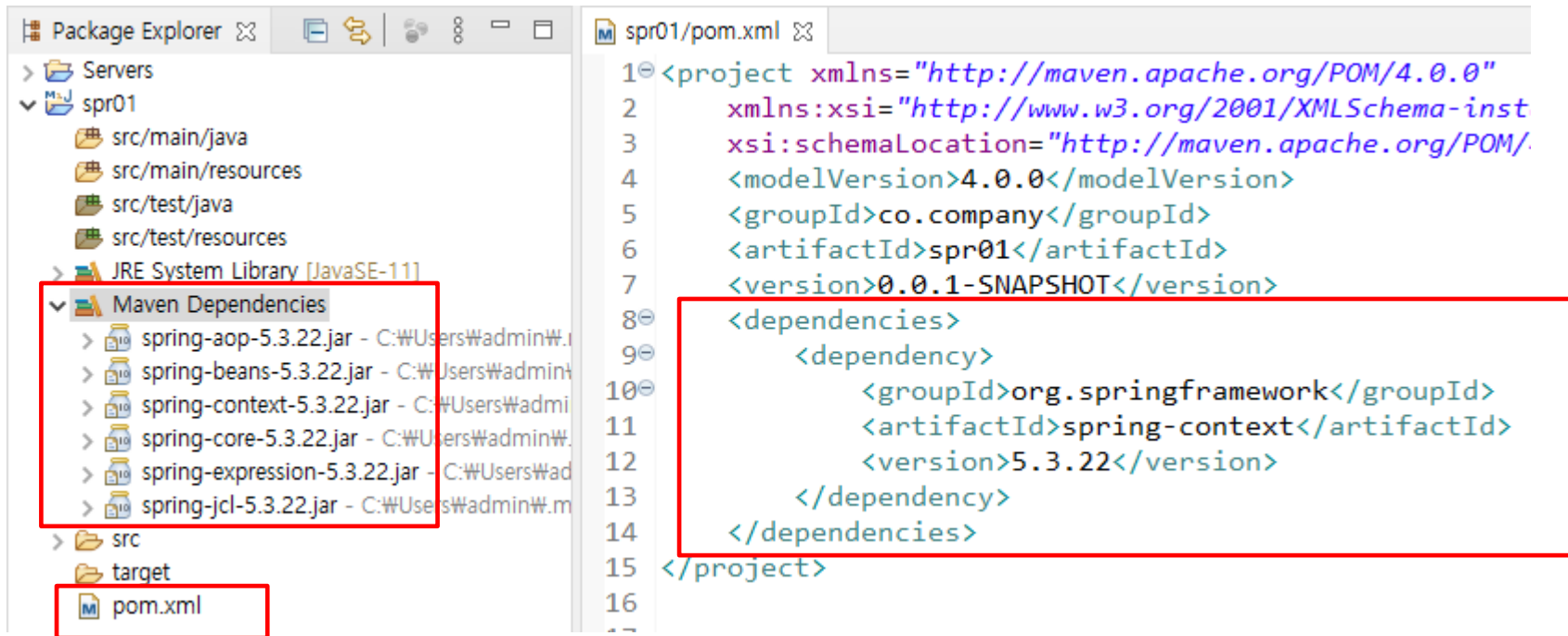
- 2. resources 폴더 선택하고 OK 버튼



## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

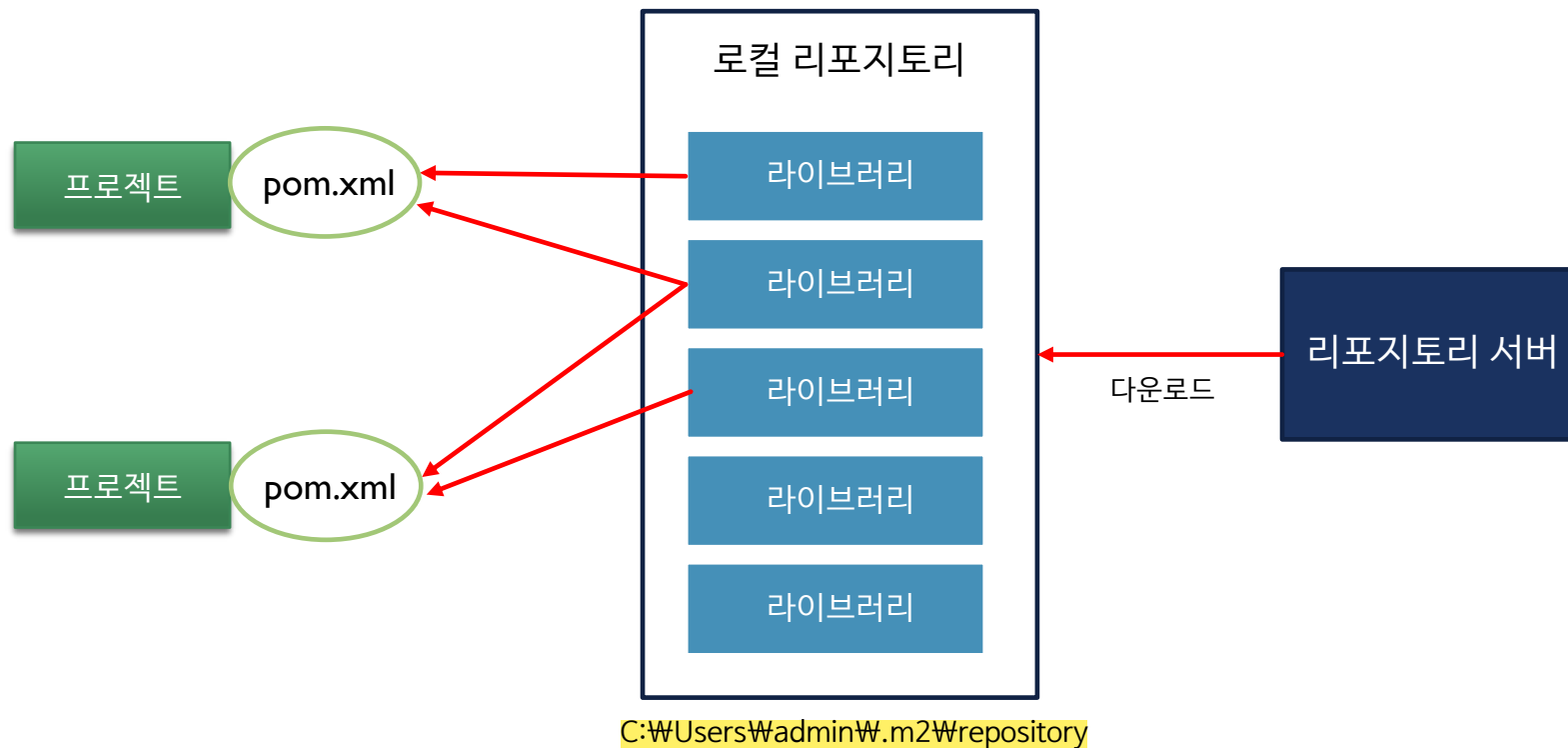
### ■ 라이브러리 의존성 관리

- 라이브러리 다운로드 자동화. 필요한(의존성 있는) 라이브러리를 하나씩 다운로드 받을 필요가 없다.
- 중앙 repository 서버에서 필요한 jar파일을 다운받아 의존 모듈을 관리한다.



## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

- 라이브러리 의존성 관리

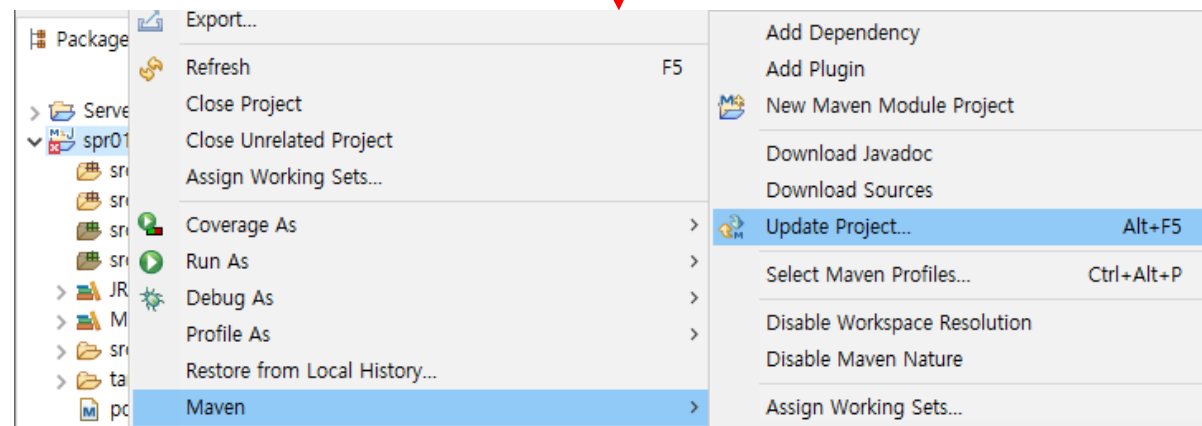
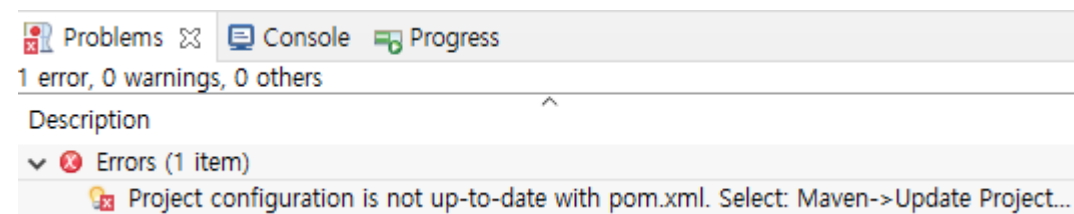




## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

### ■ 빌드 설정

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.1</version>
      <configuration>
        <source>11</source>
        <target>11</target>
        <encoding>utf-8</encoding>
        <compilerArgument>-Xlint:all</compilerArgument>
        <showWarnings>true</showWarnings>
        <showDeprecation>true</showDeprecation>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```



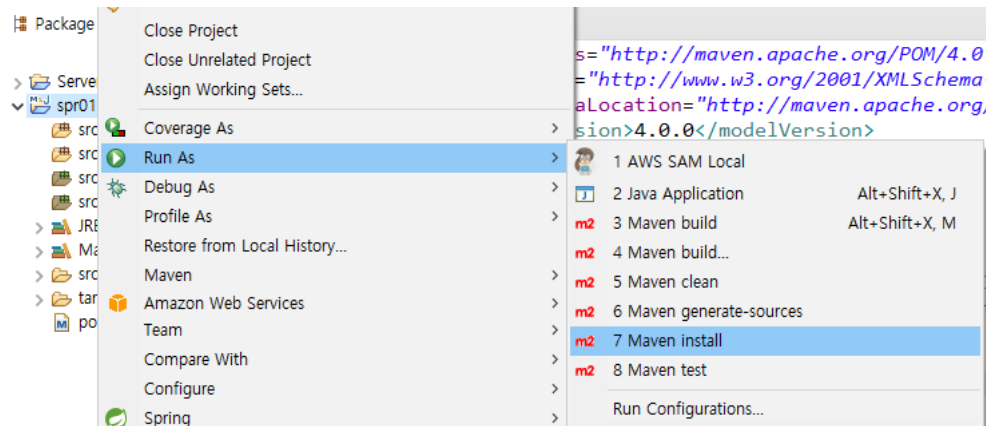
프로젝트 컨텍스트 메뉴

-> Maven 메뉴

-> Update Project... 메뉴 실행

## 3.2 Maven 프로젝트로 스프링 프로젝트 만들기

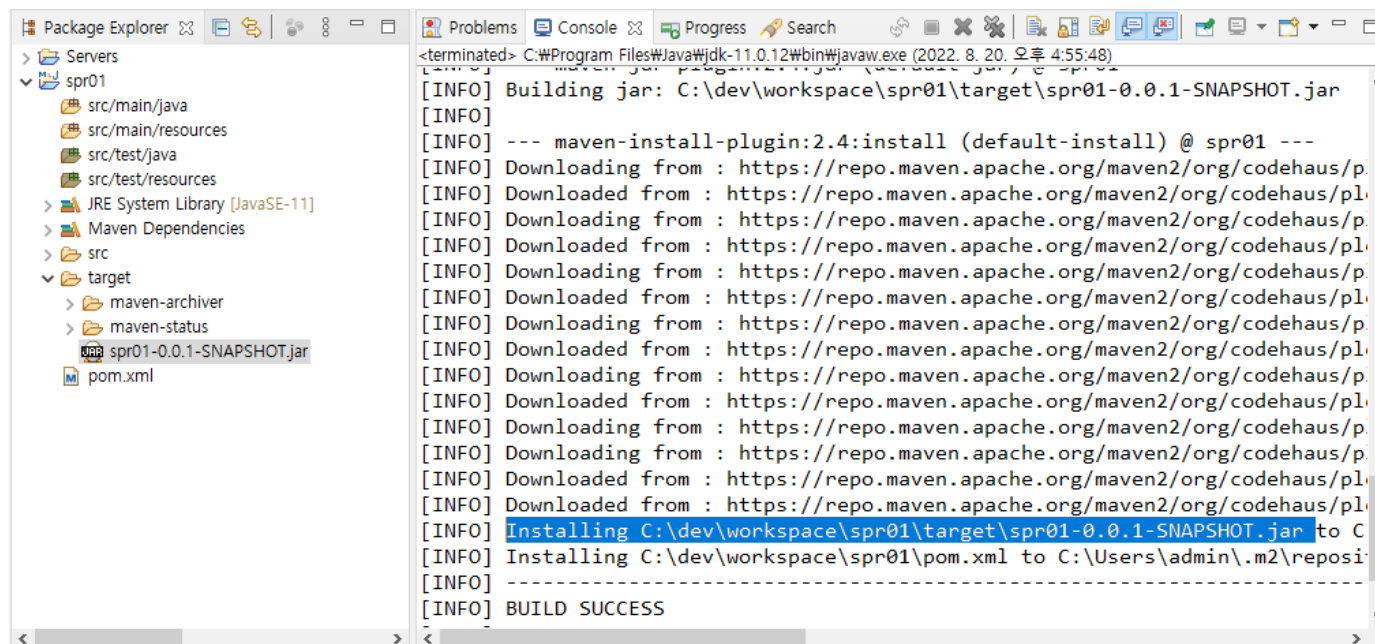
### ■ 빌드하기



프로젝트 컨텍스트 메뉴

-> Run As 메뉴

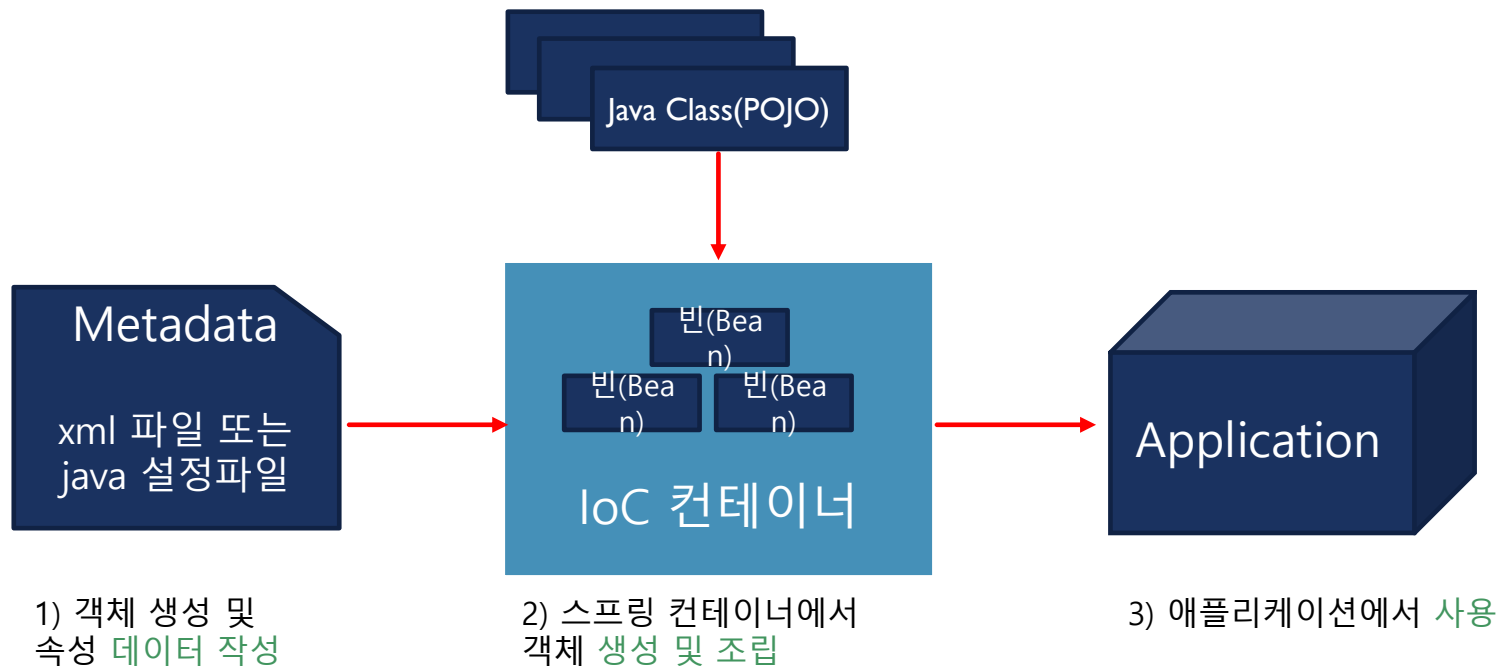
-> maven install 메뉴 실행



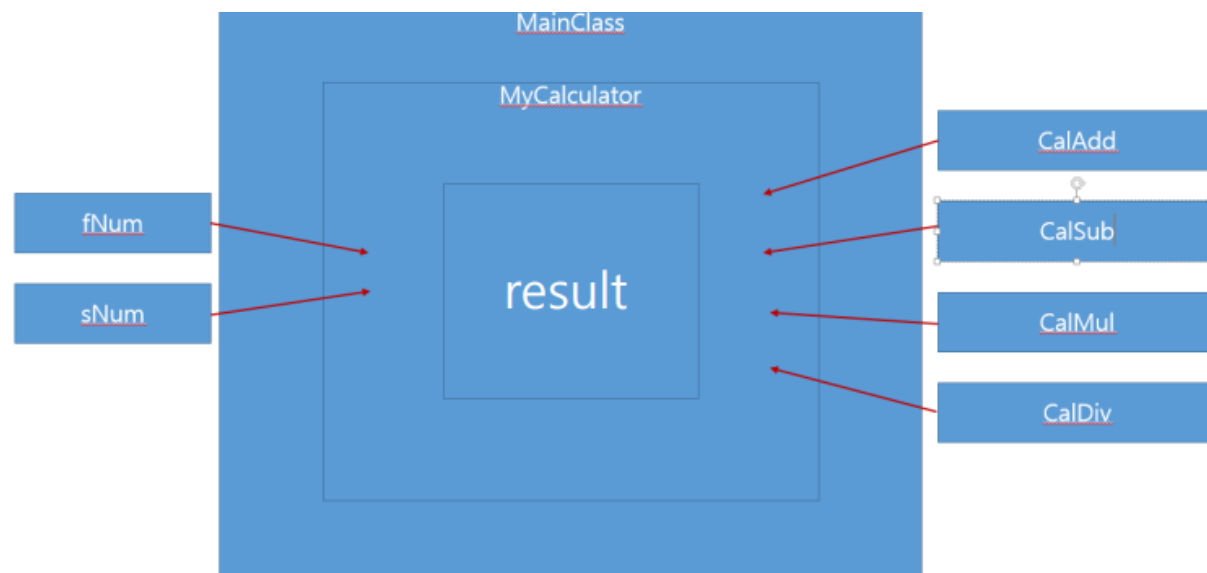
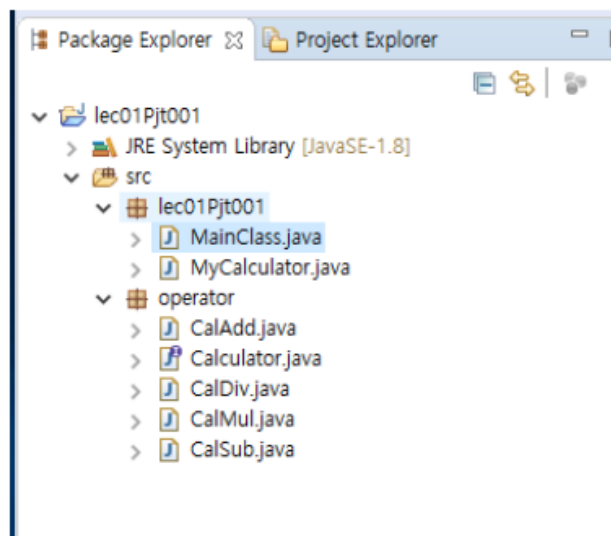
target 폴더에 배포파일(jar 또는 war) 파일이 생성됨

## 4.1 스프링 IoC 컨테이너

- ApplicationContext = 컨테이너
- 컨테이너에서 객체를 생성하고 조립(주입)해주는 역할
- 컨테이너가 관리하는 객체를 **빈(Bean)**이라고 함.



## 4.1 스프링 IoC 컨테이너



## 4.2 스프링 IoC 컨테이너 실습

## 4.2 스프링 IoC 컨테이너 실습

- 스프링 IoC 컨테이너 실습

```
public class Tv {  
    public void on() {  
        System.out.println("on");  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        SamsungTV tv = new SamsungTV();  
        tv.on();  
    }  
}
```

## 4.2 스프링 IoC 컨테이너 실습

### ■ spring 설정파일

- File -> New -> Spring Bean Configuration File
- filename : applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
<bean id="tv" class="co.spring.SamsungTV"/>
```

```
</beans>
```

```
public class Test {
    public static void main(String[] args) {
        GenericXmlApplicationContext ctx =
            new GenericXmlApplicationContext("classpath:applicationContext.xml");
        Tv tv = ctx.getBean(SamsungTV.class);
        tv.on();
    }
}
```