

# 메이븐(Maven)이란?

JAVA로 개발된 프로젝트의 전체적인 라이프사이클 관리 도구이다. 프로젝트 객체 모델(Project Object Model: POM)이라는 개념을 바탕으로 프로젝트의 의존성관리, 라이브러리관리, 프로젝트 라이프사이클 관리 기능등을 제공하는 프로젝트 관리도구이다. 플러그인을 기반으로 소스코드로부터 배포 가능한 산출물을 만들어내는 빌드(build) 기능을 제공한다.

빌드(build)란 작성한 소스코드, 프로젝트에서 쓰인 각각의 파일(.xml, .jar, ...) 등을 JVM이나 WAS가 인식할 수 있는 구조로 패키징 하는 과정 및 결과물이다.

플러그인이란 웹 브라우저의 일부 기능들을 쉽게 설치하여 사용하게 할 수 있는 프로그램을 말한다.

# 메이븐(Maven) 특징

중앙 저장소를 통한 자동 의존성 관리를 한다.

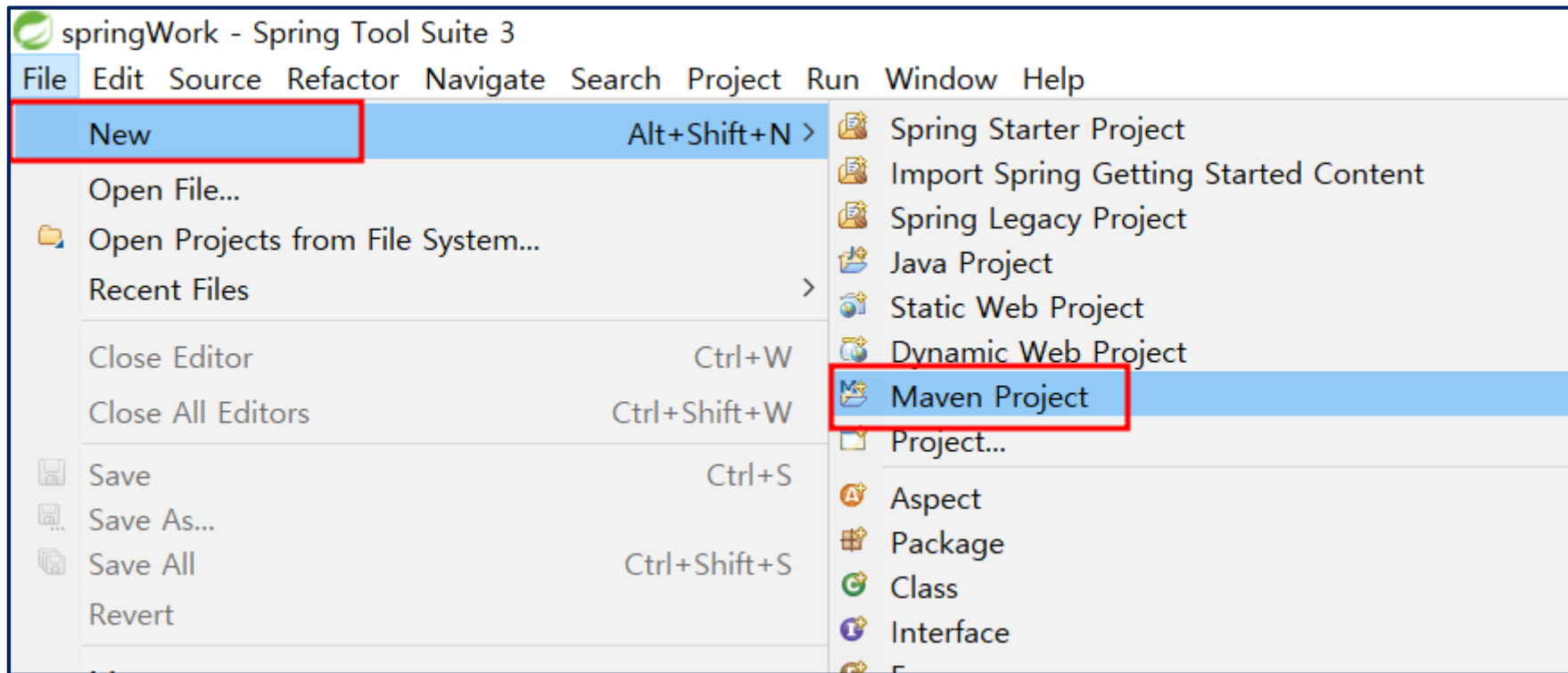
메이븐 프로젝트는 (pom.xml) 에 있는 정보를 메이븐 리파지토리 사이트에서 다운받아 jar 파일을 C: > 사용자 > 컴퓨터계정 > .m2 > repository 폴더에 저장한다. 라이브러리를 (pom.xml) 에 정의만 해주면 내가 사용할 라이브러리 뿐만 아니라 해당 라이브러리 작동에 필요한 하위 라이브러리까지 네트워크를 통해서 다운 받아준다.

정해진 빌드 방법을 사용하여 협업에서 유리하게 작용한다.

예를들어 어떠한 라이브러리를 사용할때 A 개발자는 1.1버전을 쓰고 B 개발자는 3.4버전을 쓰고 C 개발자는 5.4 버전을 쓴다면... 프로그램이 제대로 작동되지 않는다. 이렇게 되지 않도록 정해진 기준으로 라이브러리를 사용할 수 있게 한다.

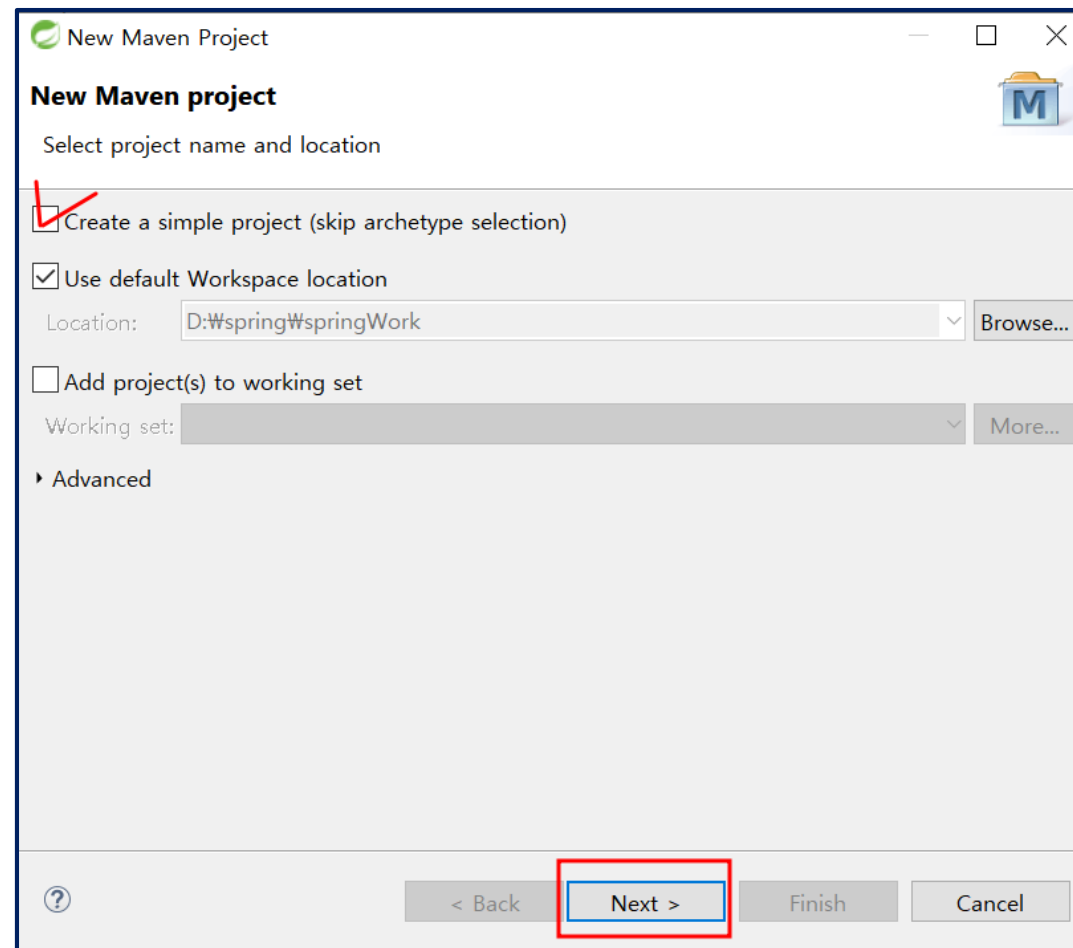
# 메이븐 프로젝트 생성하기

File > New > Maven Project 클릭!



# 메이븐 프로젝트 생성하기

Create a simple project 체크 후 Next 클릭!

A screenshot of the "New Maven Project" dialog box in an IDE. The dialog has a title bar with a green icon and the text "New Maven Project". Below the title bar, it says "New Maven project" and "Select project name and location". There are four checkboxes: "Create a simple project (skip archetype selection)" which is checked with a red checkmark, "Use default Workspace location" which is checked, "Add project(s) to working set" which is unchecked, and "Advanced" which is collapsed. Below the checkboxes, there is a "Location:" label, a text box containing "D:\spring\springWork", a "Browse..." button, and a "Working set:" label with a dropdown menu and a "More..." button. At the bottom, there are four buttons: "< Back", "Next >" (which is highlighted with a red rectangle), "Finish", and "Cancel".

New Maven Project

New Maven project

Select project name and location

☒ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location: D:\spring\springWork Browse...

☐ Add project(s) to working set

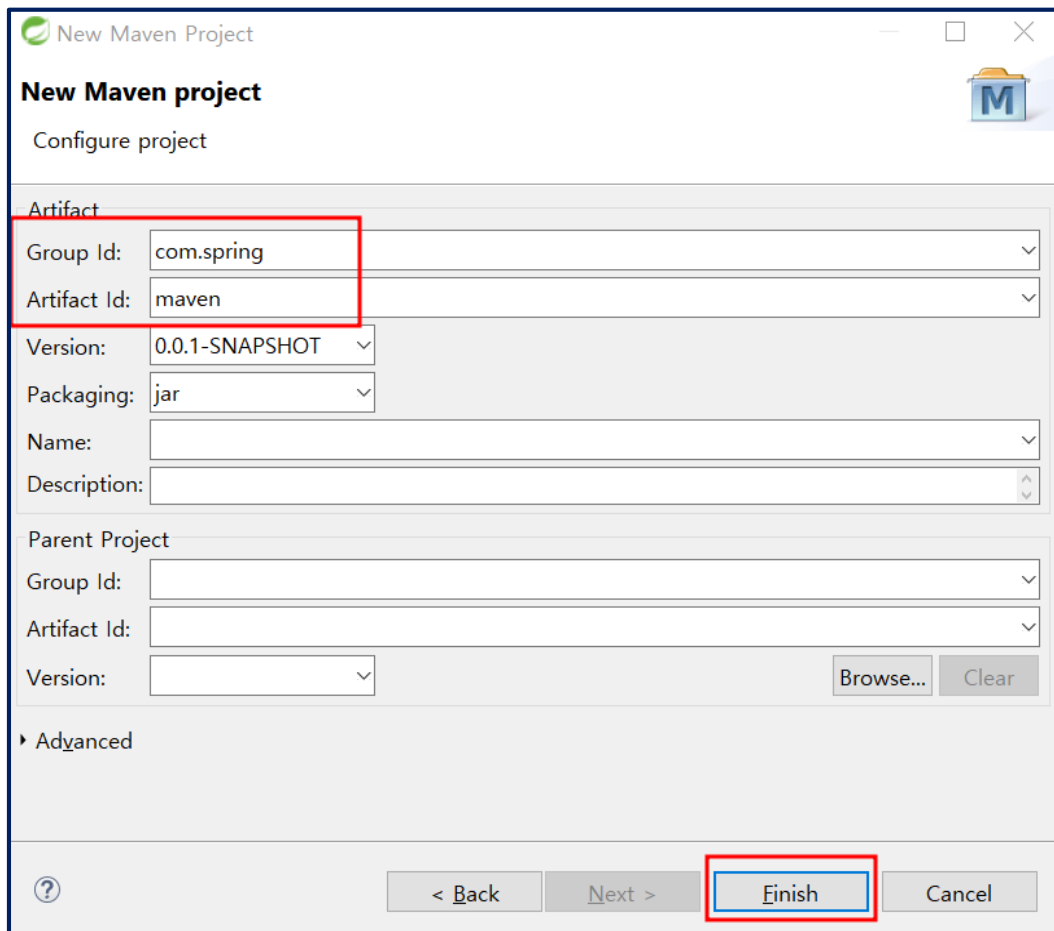
Working set: More...

Advanced

< Back Next > Finish Cancel

# 메이븐 프로젝트 생성하기

Group Id 와 Artifact Id 작성 후 Finish 클릭!



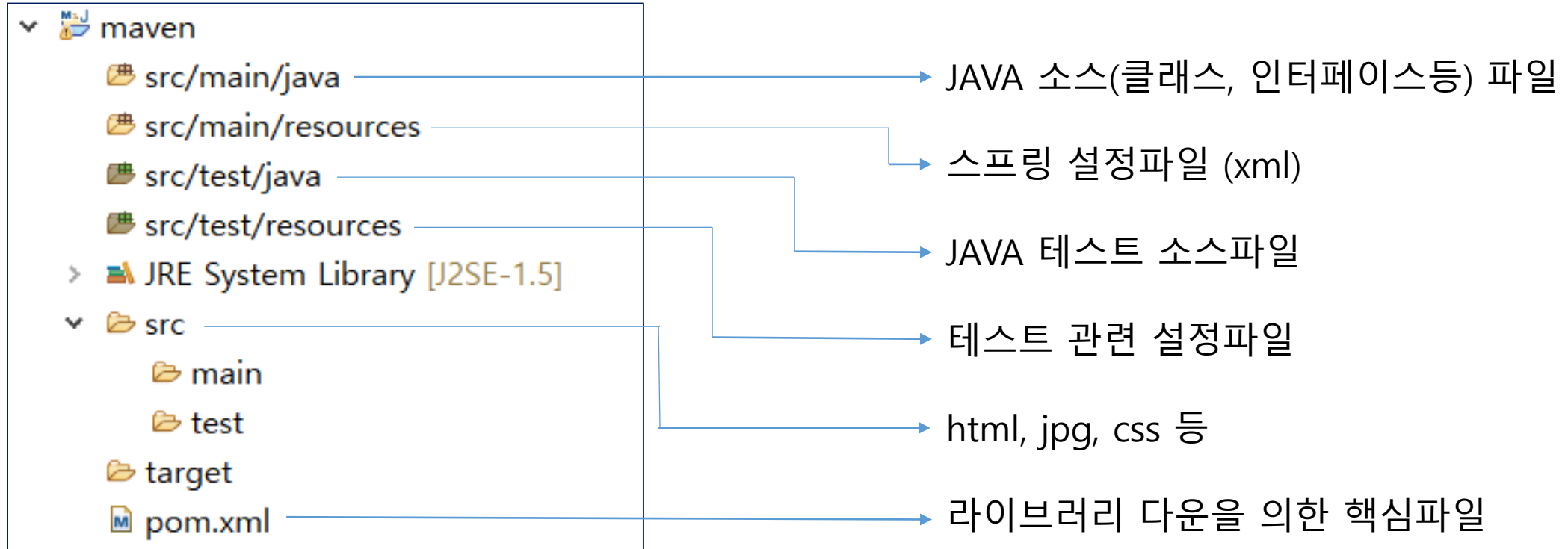
The image shows the 'New Maven Project' dialog box in an IDE. The 'Artifact' section is highlighted with a red box, showing 'Group Id: com.spring' and 'Artifact Id: maven'. The 'Parent Project' section is also visible, with empty fields for 'Group Id', 'Artifact Id', and 'Version'. At the bottom, the 'Finish' button is highlighted with a red box.

GroupId : 프로젝트를 식별할수 있는 고유한 이름.  
규칙 : 일반적으로 회사 도메인을 거꾸로 한다.

ArtifactId : ArtifactId는 프로젝트의 이름으로 사용된다. artifact는 소프트웨어의 산출물을 의미한다.  
빌드를 완료하면 .jar 파일이 만들어지고 이를 아티팩트라고 부른다.

규칙 : 특수문자를 작성하지 않고 소문자로만 작성

# 메이븐 프로젝트 구조



# pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>com.spring</groupId>
    <artifactId>maven</artifactId>
    <version>0.0.1-SNAPSHOT</version>

</project>
```

pom.xml 은 <project>와 </project> 태그 사이에 모든 정보를 기술한다.

<modelVersion>4.0.0</modelVersion>: POM모델 버전이다.

groupId / artifactId / version : 기본적으로 프로젝트를 생성할 때 입력하고 설정했던 속성값들이 pom.xml에 등록된다.

# Spring Framework 모듈

스프링 프레임워크에서 제공하고있는 모듈을 사용하려면, XML 파일등을 사용해서 모듈에 대한 설정을 해야한다.

스프링 모듈	기능
spring-core	스프링의 IoC 와 DI 기능을 제공한다.
spring-aop	관점지향 프로그래밍을 할 수 있도록 지원하는 기능을 제공한다.
spring-jdbc	데이터베이스를 쉽게 다룰 수 있는 기능을 제공한다.
spring-tx	스프링에서 제공하는 트랜잭션 기능을 제공한다.
spring-webmvc	스프링에서 제공하는 스프링 MVC 구현기능을 제공한다.





# pom.xml

<https://mvnrepository.com> > 사이트에서 검색 후에 사용할 라이브러리를 복사 후 `<dependencies></dependencies>` 태그 안에 의존 라이브러리를 등록한다.

```
<dependencies>
  <!-- spring-context 모듈 -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.18</version>
  </dependency>
</dependencies>
```

# pom.xml

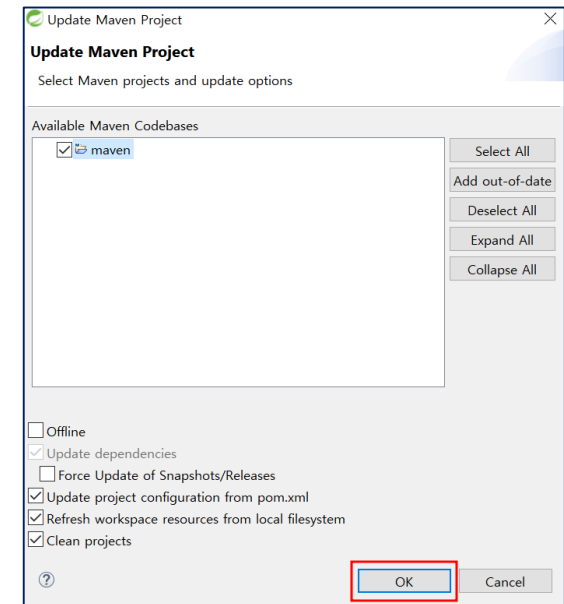
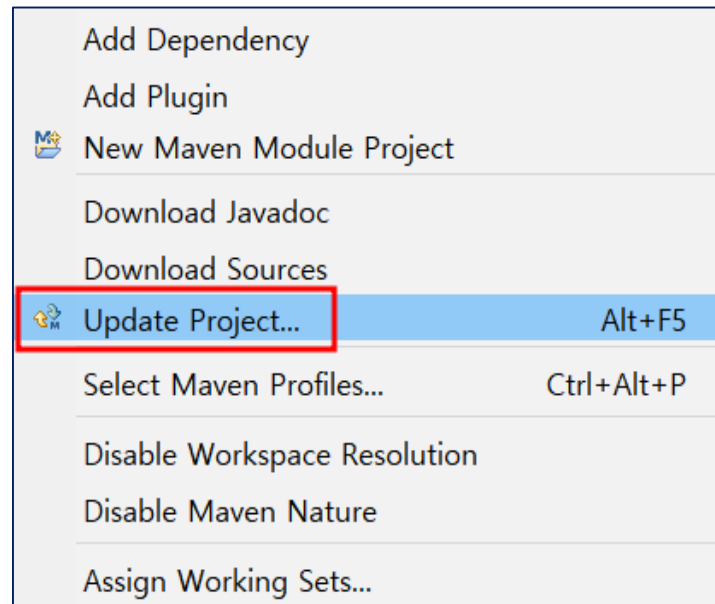
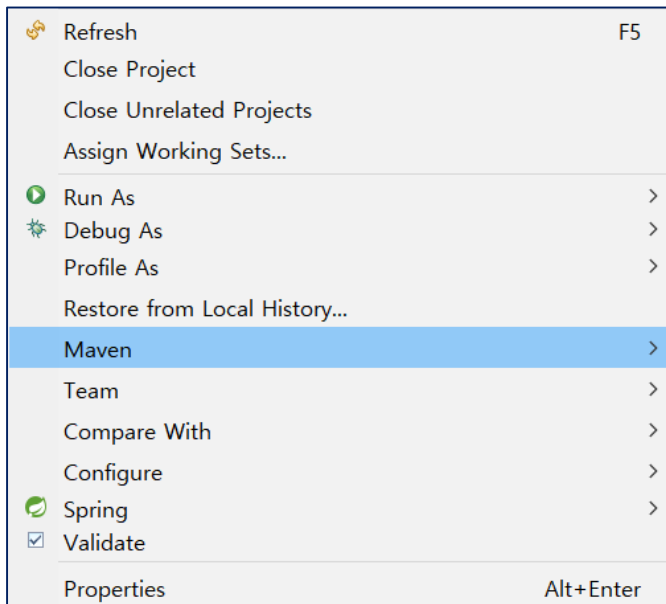
메이븐은 플러그인을 구동해주는 프레임워크(plugin execution framework)이다. 모든 작업은 플러그인에서 수행한다. 아래와 같이 설정 후에 저장을 하자.

```
<!-- 메이븐 컴파일러와 자바버전 설정 -->
<build>
  <plugins>
    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>3.8.1</version><!-- 메이븐 컴파일러를 3.8.1을 사용 -->
      <configuration>
        <source>11</source><!-- 자바의 컴파일러를 11버전 사용 -->
        <target>11</target>
        <encoding>UTF-8</encoding>
      </configuration>
    </plugin>
  </plugins>
</build>
```

# pom.xml

설정을 마쳤으면 pom.xml 에 정의한 라이브러리들을 프로젝트에 적용하기 위해 프로젝트 업데이트를 해주어야 한다.

프로젝트 우클릭 > Maven > Update Project 클릭 > 프로젝트 선택 후 OK 클릭!!



# pom.xml

Library와 Dependencies가 pom.xml에 설정한대로 적용된걸 확인 할 수 있다.

