

# Apache Maven

# ▶ Apache Maven

## ✓ Maven이란 ??

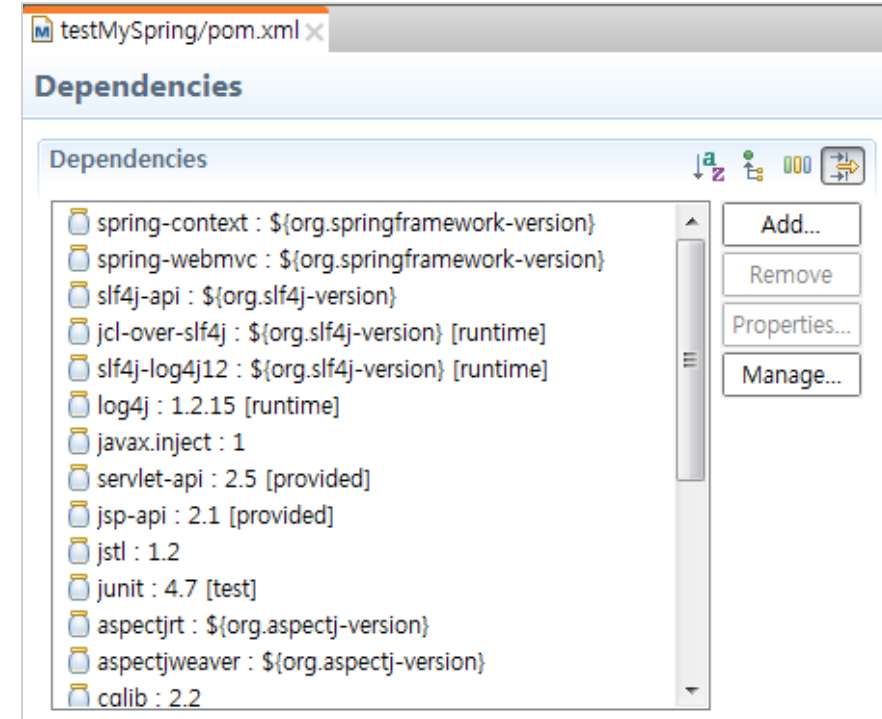
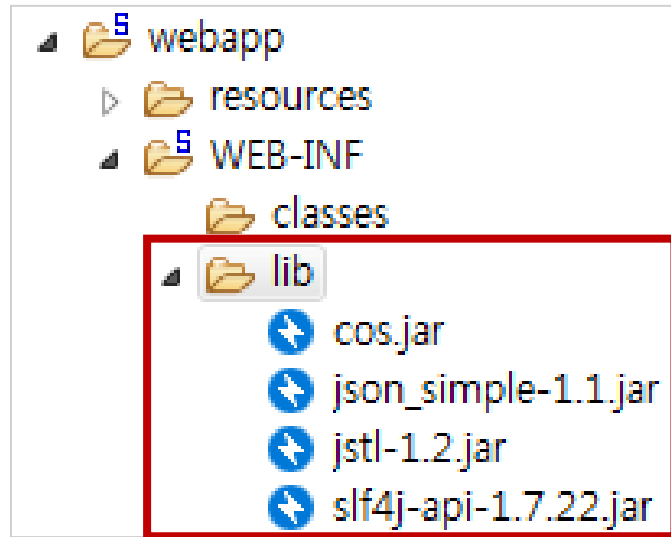
Maven이란 자바용 프로젝트 관리 도구로, project object model(POM) XML 문서를 통해 해당 프로젝트의 버전 정보 및 라이브러리 정보들을 통합하여 관리하는 프레임워크이다.

## ✓ 라이브러리 종속성

일반적인 프로젝트는 개발자가 필요한 라이브러리를 직접 찾아서 추가해야 하지만, Maven을 사용하면 pom.xml 문서에 사용하고자 하는 라이브러리를 등록하여 자동으로 프로젝트에 추가되게 하여 라이브러리 관리의 편리성을 제공해준다.

# ▶ Apache Maven

## ✓ Maven의 종속성



- \* 이제까지 프로젝트 내 lib 폴더에 사용할 라이브러리를 직접 추가하여 관리 해왔다면, Maven은 pom.xml 문서 하나만으로 필요한 라이브러리를 자동으로 설치하고 관리할 수 있다.

# ▶ Apache Maven

## ✓ POM이란 ??

POM(Project Object Model)은 하나의 프로젝트에서 사용하는 자바 버전, 라이브러리, 플러그인 구성을 통합하여 관리할 수 있게 각 설정 정보를 XML로 문서화 한 것을 말한다.

\*\* 라이브러리 관련 정보 제공 사이트 : <https://mvnrepository.com/>

# ▶ Apache Maven

## ✓ pom.xml의 구성

```
<project>
  <modelVersion>4.0.0</modelVersion> <!-- Maven 2 버전 이후 POM의 경우 항상 4.0.0 -->
  <groupId>com.kh</groupId> <!-- 최초 만든 패키지 1, 2레벨 -->
  <artifactId>spring</artifactId> <!-- 최초 만든 패키지 3레벨 = context-path -->
  <name>springProject</name> <!-- 프로젝트 명 -->
  <version>1.0</version>

  <dependencies> <!-- 라이브러리 의존성 주입 -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope> <!-- 이 라이브러리는 테스트 실행 및 컴파일에만 사용 -->
    </dependency>
  </dependencies>
</project>
```

# ▶ Apache Maven

## ✓ pom.xml의 구성

```
<project>
  <modelVersion>4.0.0</modelVersion> <!-- Maven 2 버전 이후 POM의 경우 항상 4.0.0 -->
  <groupId>com.kh</groupId> <!-- 최초 만든 패키지 1, 2레벨 -->
  <artifactId>spring</artifactId> <!-- 최초 만든 패키지 3레벨 = context-path -->
  <name>springProject</name> <!-- 프로젝트 명 -->
  <version>1.0</version>

  <dependencies> <!-- 라이브러리 의존성 주입 -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope> <!-- 이 라이브러리는 테스트 실행 및 컴파일에만 사용 -->
    </dependency>
  </dependencies>
</project>
```

# ▶ Apache Maven

## ✓ pom.xml의 구성

```
<project>
  <modelVersion>4.0.0</modelVersion> <!-- Maven 2 버전 이후 POM의 경우 항상 4.0.0 -->
  <groupId>com.kh</groupId> <!-- 최초 만든 패키지 1, 2레벨 -->
  <artifactId>spring</artifactId> <!-- 최초 만든 패키지 3레벨 = context-path -->
  <name>springProject</name> <!-- 프로젝트 명 -->
  <version>1.0</version>

  <dependencies> <!-- 라이브러리 의존성 주입 -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope> <!-- 이 라이브러리는 테스트 실행 및 컴파일에만 사용 -->
    </dependency>
  </dependencies>
</project>
```

# ▶ Apache Maven

## ✓ pom.xml의 구성

```
<project>
  <modelVersion>4.0.0</modelVersion> <!-- Maven 2 버전 이후 POM의 경우 항상 4.0.0 -->
  <groupId>com.kh</groupId> <!-- 최초 만든 패키지 1, 2레벨 -->
  <artifactId>spring</artifactId> <!-- 최초 만든 패키지 3레벨 = context-path -->
  <name>springProject</name> <!-- 프로젝트 명 -->
  <version>1.0</version>

  <dependencies> <!-- 라이브러리 의존성 주입 -->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope> <!-- 이 라이브러리는 테스트 실행 및 컴파일에만 사용 -->
    </dependency>
  </dependencies>
</project>
```



# ▶ Framework

## ✓ Framework의 특징

1. 개발자가 따라야 하는 가이드를 제공한다.
2. 개발할 수 있는 범위가 정해져 있다.
3. 개발자를 위한 다양한 도구, 플러그인들을 지원한다.

## ✓ Framework의 특징

### 장 점

- 개발 시간을 줄일 수 있음
- 정형화 되어 있어 일정수준 이상의 품질을 기대할 수 있음
- 유지 보수가 쉬움

### 단 점

- 너무 의존하면 개발자들의 능력이 떨어져서 스스로 직접 개발하는 것이 어려워짐
- 습득에 걸리는 시간이 오래 걸림

# ▶ Framework

## ✓ Framework의 종류

구 분	설 명	종류
영속성 Framework	데이터의 저장, 조회, 변경, 삭제를 다루는 클래스 및 설정 파일들을 라이브러리화하여 구현한 프레임워크	- Mybatis - Hibernate
자바 Framework	Java EE를 통한 웹 어플리케이션 개발에 초점을 맞추어 필요한 요소들을 모듈화 하여 제공하는 프레임워크	- Spring Framework - 전자정부표준 - Spring - Struts
화면 구현 Framework	Front-End를 보다 쉽게 구현할 수 있게 틀을 제공하는 프레임워크	- Bootstrap - Foundation - MDL
기능 및 지원 Framework	특정 기능이나 업무 수행에 도움을 줄 수 있는 기능을 제공하는 프레임워크	- Log4j - JUnit 5 - ANT