



Today's lecture



```
System.out.print('Apache Maven');
```

▶ Apache Maven

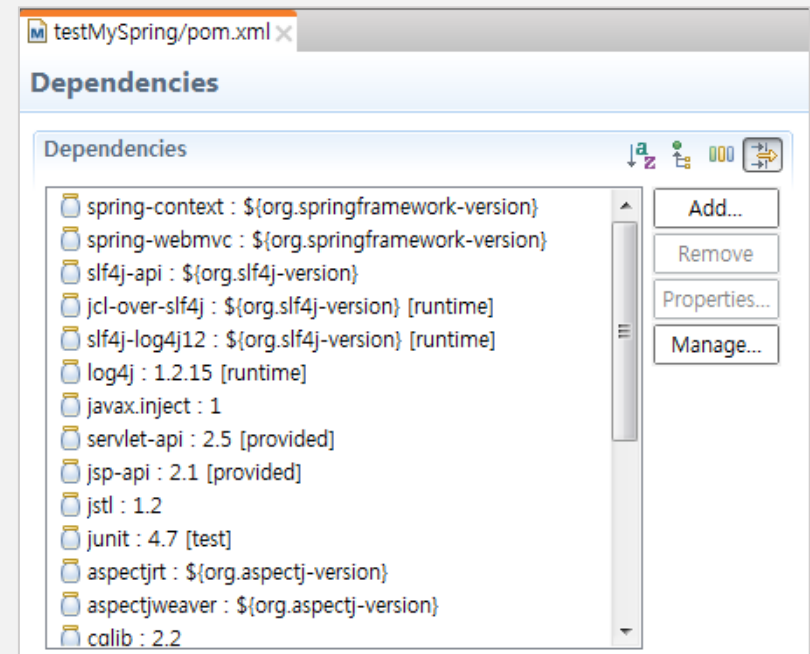
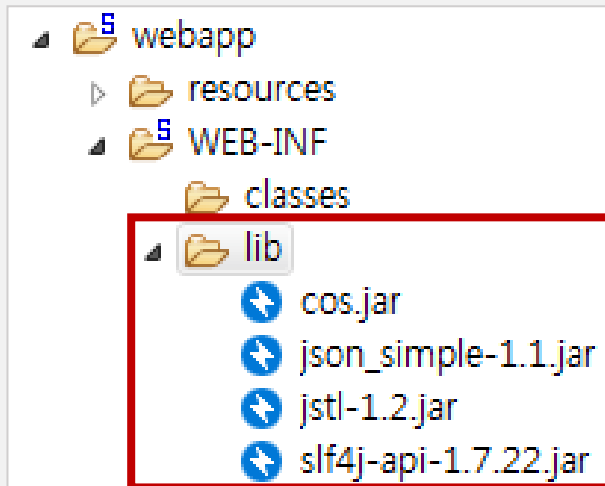
자바용 프로젝트 관리 도구로 Project Object Model(POM) XML문서를 통해 해당 프로젝트의 버전 정보 및 라이브러리 정보들을 통합하여 관리하는 프레임워크

✓ 라이브러리 종속성

일반적인 프로젝트는 개발자가 필요한 라이브러리를 직접 찾아서 추가해야 하지만 Maven을 사용하면 pom.xml문서에 사용하고 싶은 라이브러리를 등록하여 자동으로 프로젝트에 추가되게 하여 라이브러리 관리의 편리성 제공

▶ 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 있음

▶ POM

POM(Project Object Model)은 하나의 프로젝트에서 사용하는 자바 버전,
라이브러리, 플러그인 구성을 통합하여 관리할 수 있게
각 설정 정보를 XML로 문서화한 것

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

▶ 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>
```