

# Apache Maven



## Apache Maven에 대해서

Apache Maven은 Project Management 도구입니다. Apache Ant를 빌드를 위해서 만들어졌지만 Apache Maven은 프로젝트를 관리하기 위해 만들어졌습니다. Maven 안에는 소스코드 빌드, 테스트, 릴리즈 등의 다양한 기능을 포함하고 있습니다.

blocked URL

- Apache Maven은 프로젝트 관리 도구이다.
- Project를 Model로 표현한다. 이것을 POM(Project Object Model)이라고 하며 pom.xml 파일이 POM이다.
- Apache Maven은 Dependency Management를 통해 Dependency를 추가하고 관리할 수 있다.
- Project의 Lifecycle Management를 통해 소스코드 빌드, 테스트, 배포, 릴리즈 등의 기능을 제공한다.
- POM에 Plugin을 추가하여 Maven의 기능을 확장할 수 있다.
- Maven을 이해하기 위한 핵심은 Lifecycle Management, Dependency Management이다.
- Maven에서 가장 어려운 것이 Dependency Management이다.
- Maven은 프로젝트의 디렉토리 구조를 표준화하여 제공한다.

## 이 문서 검색

### 인기있는 주제

- [maven](#)
- [메이븐](#)
- [plugin](#)
- [플러그인](#)
- [drawio](#)
- [addjars](#)
- [dependency](#)
- [jar](#)
- [repository](#)
- [wrapper](#)

## 페이지

- [시작하기](#)
- [프로젝트가 시작되면 가장 먼저 하는 일](#)
- [Maven의 유래와 목표](#)
- [Maven POM의 기본 구조](#)
- [Maven Project 구조](#)
- [Maven 설치](#)
- [Maven Lifecycle](#)
- [Maven Repository](#)
- [Maven Dependency](#)
- [Maven Sub Module 및 Dependency](#)
- [Maven Deploy](#)
- [기타](#)
  - [Archetype 기반으로 프로젝트 생성하기](#)
  - [Maven Repository Mirror 설정](#)
  - [Maven Wrapper](#)
- [주요 플러그인](#)
  - [addjars Plugin](#)
    - [Maven Plugin 개발하기](#)
  - [Shade Plugin](#)

## 최근에 변경된 페이지

[Maven Sub Module 및 Dependency](#)  
10월 08, 2023 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Deploy](#)  
1월 31, 2021 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Deploy](#)  
1월 30, 2021 • 업데이트 by [오현아](#) • [변경 보기](#)

[Maven Plugin 개발하기](#)  
5월 31, 2020 • 작성자 [Edward](#)

[Archetype 기반으로 프로젝트 생성하기](#)  
5월 05, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Repository Mirror 설정](#)  
5월 05, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Dependency](#)  
5월 05, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Lifecycle](#)  
4월 26, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)

[Maven Wrapper](#)  
4월 26, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)

[Apache Maven](#)  
4월 26, 2020 • 업데이트 by [Edward](#) • [변경 보기](#)



### JAR 파일에 MANIFEST.MF 파일 수동으로 지정하기

[Edward](#)님이 4월 26, 2020에 게시했습니다.

Maven JAR Plugin을 이용하여 별도로 MANIFEST.MF 파일을 수동으로 지정하기 위해서 다음과 같이 사용할 수 있습니다.

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <version>2.6</version>
  <configuration>
    <archive>
      <index>true</index>
      <manifestFile>src/main/resources/META-INF/MANIFEST.MF</manifestFile>
      <manifest>
        <addDefaultImplementationEntries/>
      </manifest>
    </archive>
  </configuration>
</plugin>
```

- [manifest](#)
- [maven](#)
- [jar](#)
- [plugin](#)
- [플러그인](#)



[Dependency Plugin](#)으로 [dependency .jar 파일 복사하기](#)

Edward님이 4월 26, 2020에 게시했습니다.

Maven Dependency Plugin을 이용하면 Maven PM에 추가한 dependency의 모든 .jar 파일을 빌드시 추출할 수 있습니다.

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.3</version>
  <executions>
    <execution>
      <id>copy-dependencies</id>
      <phase>package</phase>
      <goals>
        <goal>copy-dependencies</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <outputDirectory>target/lib</outputDirectory>
    <overwriteIfNewer>true</overwriteIfNewer>
  </configuration>
</plugin>
```

이 방법을 잘 이용하면 .jar 패키징시 dependency 파일들을 .jar 파일 내에 같이 패키징할 수 있습니다. 특히 이 방법은 .war 파일의 /WEB-INF/lib 디렉토리에 .jar 파일을 복사하는 개념과 동일하다고 볼 수 있습니다.

- [maven](#)
- [메이븐](#)
- [dependency](#)
- [jar](#)
- [copy](#)
- [패키징](#)
- [packaging](#)