



이클립스 버전에서 대해 아래와 같이 더 이상 Spring Tool 3이 더 이상 지원되지 않습니다.

**Spring Tools 3 (Standalone Edition) 3.9.14.RELEASE**


Spring Tools 3 (Standalone Edition) Attention - End of life: The Spring Tools 3 are no longer maintained and no longer updated for newer Eclipse versions. Please...
[more info](#)

by [VMware](#), [EPL](#)
[J2EE](#) [spring](#) [Spring IDE](#) [Cloud](#) [jee](#)

★ 110

 Installs: **351K** (5,769 last month)


Install

**Spring Tools 3 Add-On for Spring Tools 4 3.9.22.RELEASE**

Spring Tools 3 Add-On for Spring Tools 4 End of life: This Spring Tools 3 Add-On for Spring Tools 4 is no longer maintained or updated. The final and last release...
[more info](#)

by [VMware](#), [EPL](#)
[J2EE](#) [spring](#) [Spring IDE](#) [Cloud](#) [jee](#)

★ 322


 Installs: **581K** (4,338 last month)

Install


따라서 도서의 스프링 레거시 프로젝트(Spring Legacy Project) 생성을 다음과 같은 메이븐 프로젝트(Maven Project) 생성으로 변경하여 실습할 수 있도록 개발 환경설정을 변경합니다.

2.1 스프링 MVC 애플리케이션 개발 환경설정


2.1.1 자바 설치 및 환경설정하기

 [도서 참고](#)

2.1.2 웹 서버 톰캣 설치하기

 [도서 참고](#)

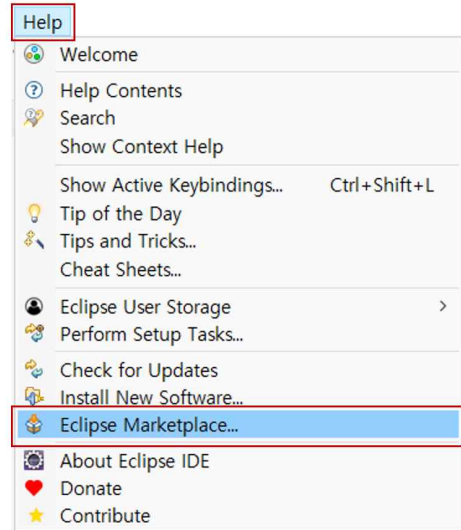
2.1.3 통합 개발 환경 이클립스 설치하기

 [도서 참고](#)

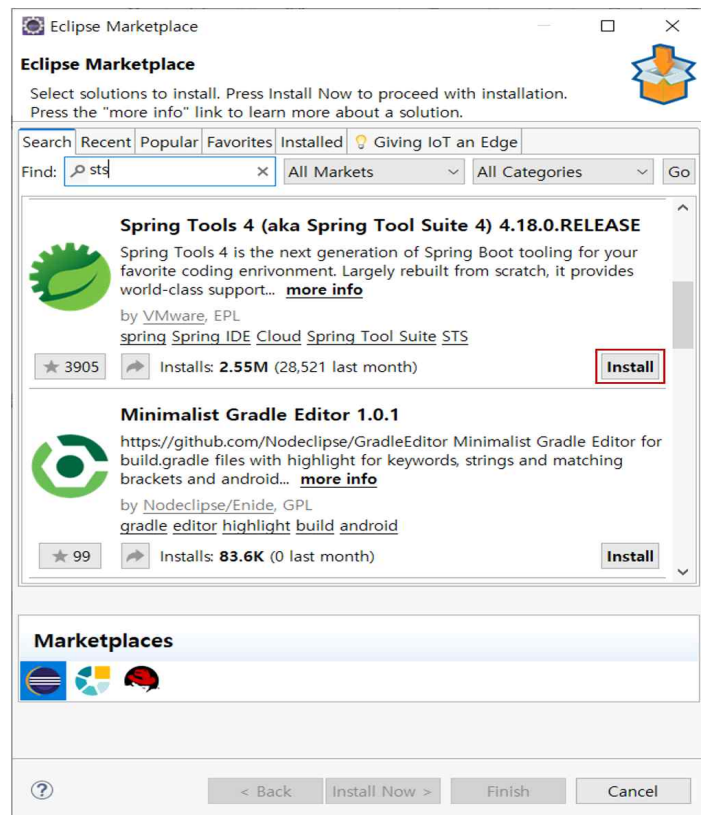
2.1.4 스프링 개발을 위한 STS 설치하기

이클립스 버전에 맞는 STS 선택하기

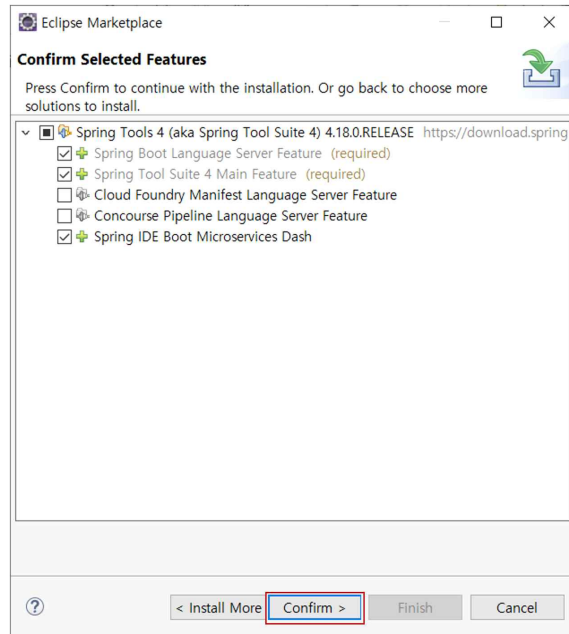
1. 이클립스의 Help → Eclipse Marketplace를 선택합니다.



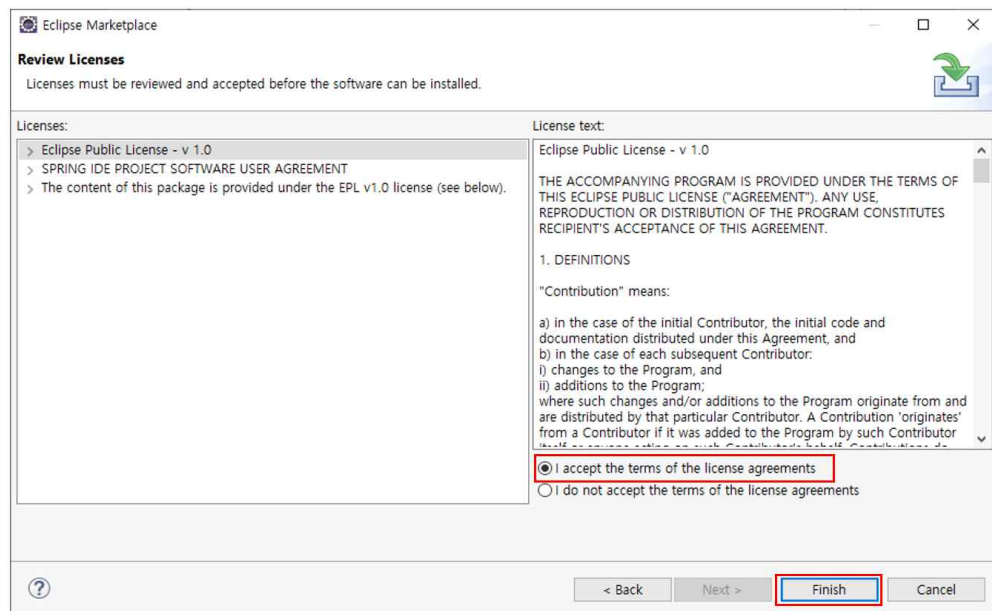
2. Eclipse Marketplace 창의 Find 항목에서 Spring Tool 4를 검색하여 자신이 사용하는 이클립스에 맞는 버전을 설치합니다. 여기에서는 Spring Tools 4(aka Spring Tool Suite 4) 4.18.0.RELEASE에 있는 Install을 클릭합니다.



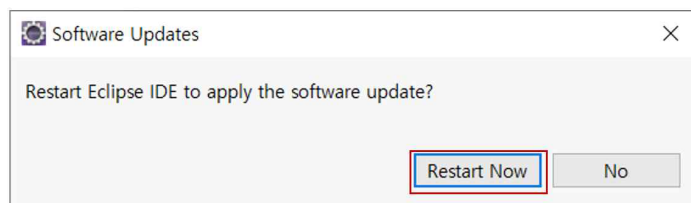
3. Eclipse Marketplace 창에 STS에 포함된 각종 기능이 나타납니다. Confirm을 클릭하여 STS를 설치합니다.



4. 소프트웨어 저작권 동의 여부를 묻는 창이 나타나면 동의한 후 Finish를 클릭합니다.



5. 설치가 완료되면 다시 시작하는지 물어보는 창이 나옵니다. Restart Now를 클릭합니다.

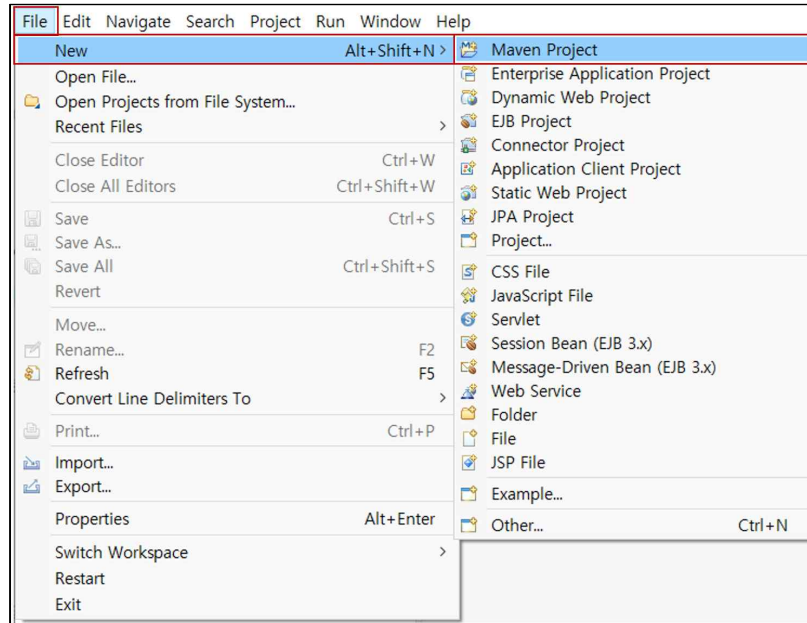


2.2 첫 번째 스프링 MVC 애플리케이션 구현

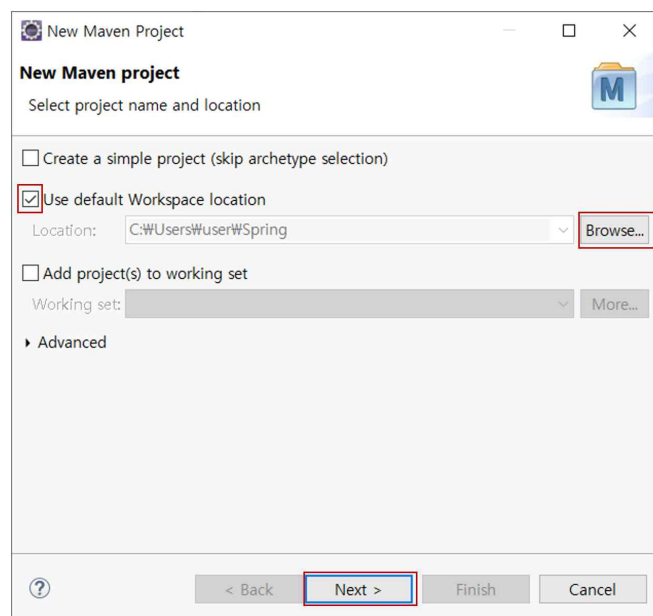
2.2.1 프로젝트 생성하기

Maven 프로젝트 생성하기

1. 이클립스에서 File → New → Project를 선택합니다.



2. New Maven Project 창에서 두 번째 체크박스를 선택하고, Browse..버튼을 클릭하여 프로젝트를 생성할 경로를 선택한 후 Next를 클릭합니다.



3. New Maven Project 창에서 Group Id는 org.apache,maven,archetype이고, Artifact Id는 maven-archetype-webapp를 선택한 후 Next를 클릭합니다.

The dialog box is titled 'New Maven Project' and 'New Maven project'. It shows 'Select an Archetype'. The 'Catalog' is set to 'Internal'. The 'Filter' is 'webapp'. A table lists available archetypes:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-webapp	1.0

Below the table, it says 'An archetype which contains a sample Maven Webapp project.' There are checkboxes for 'Show the last version of Archetype only' (checked) and 'Include snapshot archetypes' (unchecked). At the bottom, the 'Next >' button is highlighted with a red box.

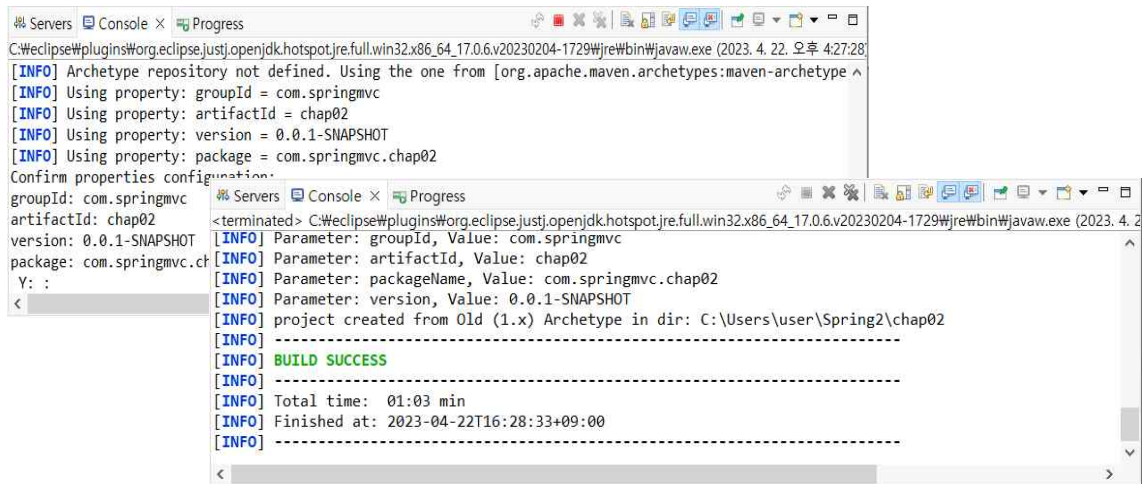
4. New Maven Project 창에서 Group Id 항목에 'com.springmvc'를 입력하고, Artifact Id 항목에 'chap02'를 입력하고 Finish를 클릭합니다.

The dialog box is titled 'New Maven Project' and 'New Maven project'. It shows 'Specify Archetype parameters'. The 'Group Id' is 'com.springmvc', 'Artifact Id' is 'chap02', 'Version' is '0.0.1-SNAPSHOT', and 'Package' is 'com.springmvc.chap02'. The checkbox 'run archetype generation interactively' is checked. Below, there is a table for 'Properties available from archetype':

Name	Value

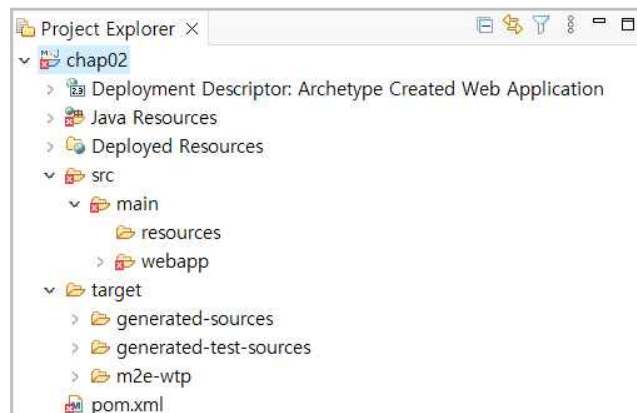
At the bottom, the 'Finish' button is highlighted with a red box.

5. 이클립스의 콘솔창에 앞에서 설정한 내용으로 프로젝트 생성을 확인하는 Y가 나타나면 Enter키를 누르고 BUILD SUCESS가 출력되면 프로젝트 생성이 완료됩니다.



```
Servers Console × Progress
C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (2023. 4. 22. 오후 4:27:28)
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart]
[INFO] Using property: groupId = com.springmvc
[INFO] Using property: artifactId = chap02
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = com.springmvc.chap02
Confirm properties configuration:
groupId: com.springmvc
artifactId: chap02
version: 0.0.1-SNAPSHOT
package: com.springmvc.chap02
Y: :
<
[INFO] Parameter: groupId, Value: com.springmvc
[INFO] Parameter: artifactId, Value: chap02
[INFO] Parameter: packageName, Value: com.springmvc.chap02
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\user\Spring2\chap02
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:03 min
[INFO] Finished at: 2023-04-22T16:28:33+09:00
[INFO] -----
```

이때 프로젝트 탐색기에 생성된 chap02 프로젝트의 구조는 오른쪽 그림과 같습니다



[NOTE] 프로젝트 오류 수정하기

1. pom.xml 오류 수정

pom.xml파일에 다음 코드를 추가합니다.

```
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <version>3.2.2</version>
</plugin>
```

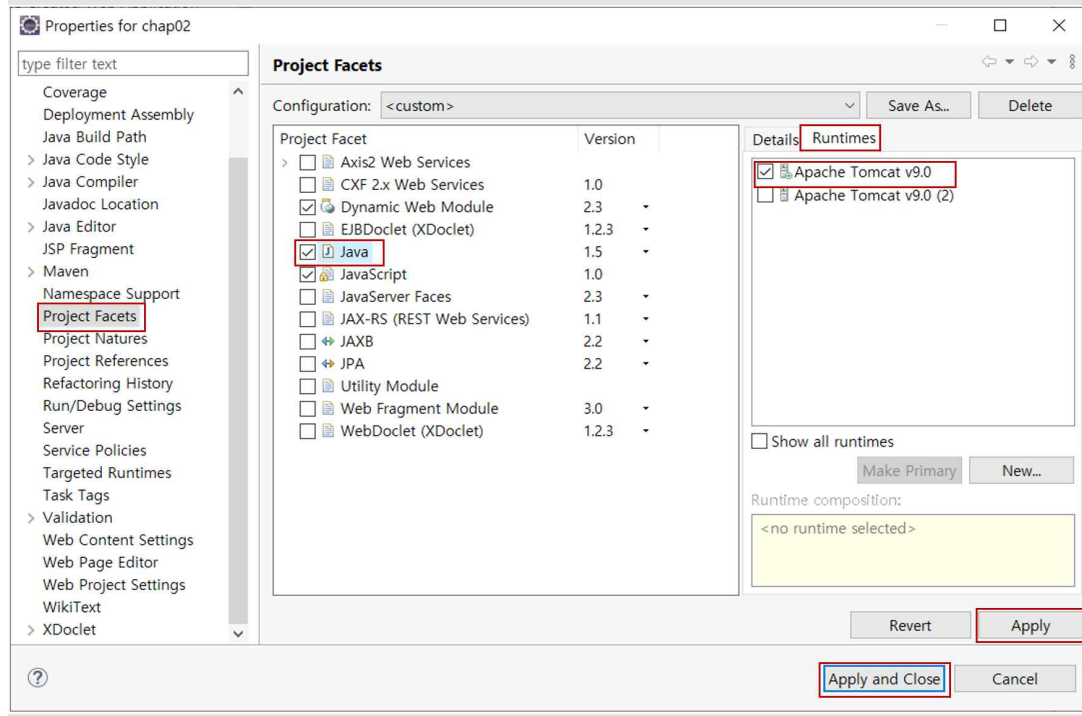
pom.xml

```
<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/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  ...

  <build>
    <finalName>chap02</finalName>
    <plugins>
      <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.2</version>
      </plugin>
    </plugins>
  </build>
</project>
```

2. webapp폴더의 index.jsp 오류 수정하기

프로젝트 우클릭 -> Properties 메뉴 선택한 후, Properties창에서 Project Facets항목에서Java -> Runtimes -> Tomcat 체크 -> Apply를 선택합니다.



코드 작성하기

본 도서에서 실습 코드를 보고 직접 따라서 작성하거나, 도서에서 제공되는 실습 코드를 복사하여 사용하여도 됩니다.

1. pom.xml

도서에서 제공하는 실습 코드를 복사하여 사용할 경우] 장별로 작성된 스프링 레거시 프로젝트의 pom.xml 코드를 복사하여 붙여넣기를 한 후 코드 아래도 이동하여 다음 코드를 추가합니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<project ...
....
    </plugin>
    <plugin>
        <groupId>org.codehaus.mojo</groupId>
        <artifactId>exec-maven-plugin</artifactId>
        <version>1.2.1</version>
        <configuration>
            <mainClass>org.test.int1.Main</mainClass>
        </configuration>
    </plugin>

    <plugin>
        <artifactId>maven-war-plugin</artifactId>
        <version>3.2.2</version>
    </plugin>
</plugins>
</build>
</project>
```

실습 코드를 보고 직접 따라서 작성할 경우] 필요한 스프링 의존라이브러리만 추가 작성합니다.

```
<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/maven-v4\_0\_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.springmvc</groupId>
    <artifactId>chap07</artifactId>
```



```

<packaging>war</packaging>
<version>0.0.1-SNAPSHOT</version>
<name>chap02 Maven Webapp</name>
<url>http://maven.apache.org</url>

<properties>
    <java-version>11</java-version>
    <org.springframework-version>5.3.19</org.springframework-version>
    <org.aspectj-version>1.9.9.1</org.aspectj-version>
    <org.slf4j-version>1.7.36</org.slf4j-version>
</properties>

<dependencies>
    <!-- Spring -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>${org.springframework-version}</version>
        <exclusions>
            <!-- Exclude Commons Logging in favor of SLF4j -->
            <exclusion>
                <groupId>commons-logging</groupId>
                <artifactId>commons-logging</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>${org.springframework-version}</version>
    </dependency>

    <!-- Logging -->
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-api</artifactId>
        <version>${org.slf4j-version}</version>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>

```

```

        <artifactId>jcl-over-slf4j</artifactId>
        <version>${org.slf4j-version}</version>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-log4j12</artifactId>
        <version>${org.slf4j-version}</version>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.15</version>
        <exclusions>
            <exclusion>
                <groupId>javax.mail</groupId>
                <artifactId>mail</artifactId>
            </exclusion>
            <exclusion>
                <groupId>javax.jms</groupId>
                <artifactId>jms</artifactId>
            </exclusion>
            <exclusion>
                <groupId>com.sun.jdmk</groupId>
                <artifactId>jmxtools</artifactId>
            </exclusion>
            <exclusion>
                <groupId>com.sun.jmx</groupId>
                <artifactId>jmxri</artifactId>
            </exclusion>
        </exclusions>
        <scope>runtime</scope>
    </dependency>
</dependency>
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
</dependency>
</dependencies>

```

```

<build>
  <finalName>chap02</finalName>
  <plugins>
    <plugin>
      <artifactId>maven-war-plugin</artifactId>
      <version>3.2.2</version>
    </plugin>
  </plugins>
</build>
</project>

```

2. web.xml

장별로 작성된 스프링 레거시 프로젝트의 web.xml 코드를 복사하여 붙여넣기를 한 후 코드를 수정합니다.

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app>

    ...

</web-app>

```

3. src/main/resources/log4j.xml

장별로 작성된 스프링 레거시 프로젝트의 resources폴더를 복사하여 붙여넣기를 한 후 log4j.xml 코드의 두 번째 줄 수정합니다.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration SYSTEM
"http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/xml/doc-files/log4j.dtd"
>
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

    ...

</log4j:configuration>

```