



!!!MVC 넘어가기 전 셋팅!!!

pom.xml에 라이브러리 버전와 값을 변경해주어야 한다.

1. 변경 사항

1) 자바 및 스프링 버전

```
<packaging>war</packaging>
<version>1.0.0-BUILD-SNAPSHOT</version>
<properties>
    <java-version>11</java-version>
    <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
    <org.aspectj-version>1.6.10</org.aspectj-version>
    <org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
<dependencies>
    <!-- spring -->
```

- <java-version> 태그 내의 자바 버전을 현재 사용 중인 버전으로 변경 → 11
- <org.springframework-version> 스프링 버전 변경 → 5.0.7

2) runtime 삭제

```

<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>

```

- 하단 <scope> 태그 변경 → 삭제 or 주석 처리

3) servlet 버전 변경

```

<!-- Servlet -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.1.0</version>
    <scope>provided</scope>
</dependency>
<dependency>

```

- servlet-api → javax.servlet-api
- 2.5 → 3.1.0

4) JUnit 버전 변경

```

<!-- Test -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>

```

- 4.7 → 4.12

5) Plugin 버전 변경

```

</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-compiler-plugin</artifactId>
    <version>2.5.1</version>
    <configuration>
        <source>11</source>
        <target>11</target>
        <compilerArgument>-Xlint:all</compilerArgument>
        <showWarnings>true</showWarnings>
        <showDeprecation>true</showDeprecation>
    </configuration>
</plugin>
<plugin>

```

- 1.6 → 11 (자바 버전과 동일)

2. 추가 - lombok

lombok 라이브러리는 JavaBean 객체를 만들거나 추가적인 작업을 간단하게 변경해주는 역할을 한다.

우선 lombok.jar 파일 위치를 파악한다.

본인은 C:/dev/jar 경로에 모든 jar를 모아놓기 때문에 해당 경로를 참조하도록 한다.

1. 룸복 설치

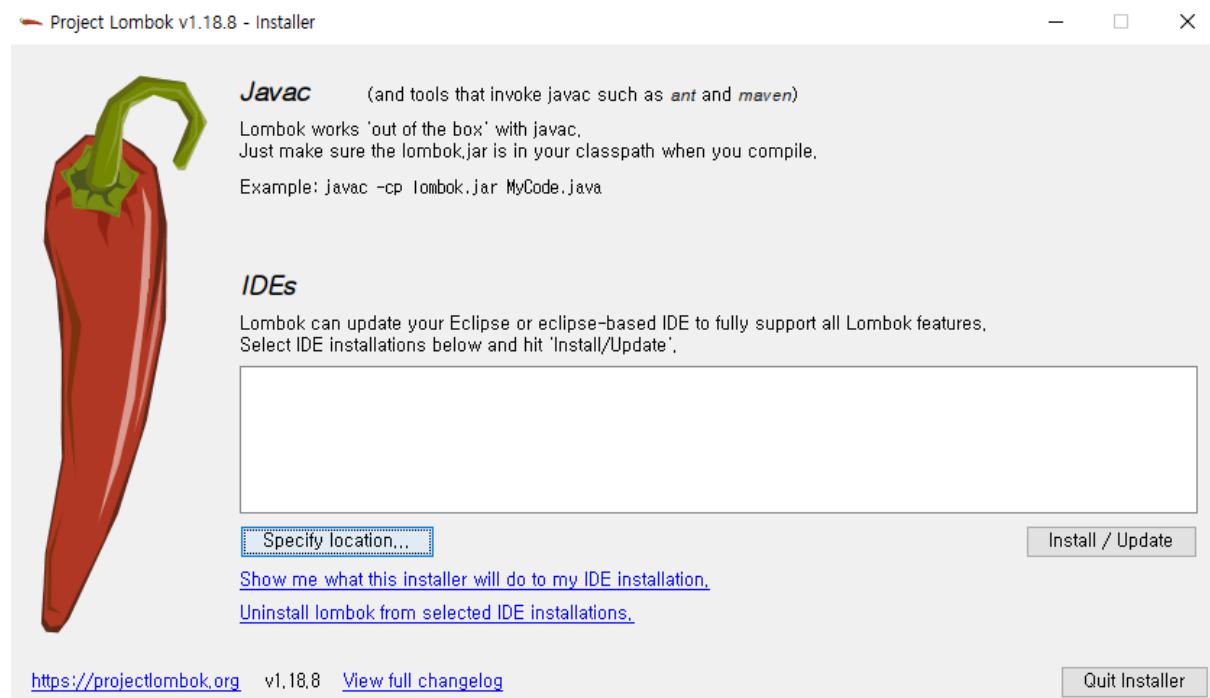
cmd 창을 이용하여 설치한다.



!! 명령어
java -jar 경로

```
C:\Users\goott1>java -jar C:\dev\jar\lombok.jar
```

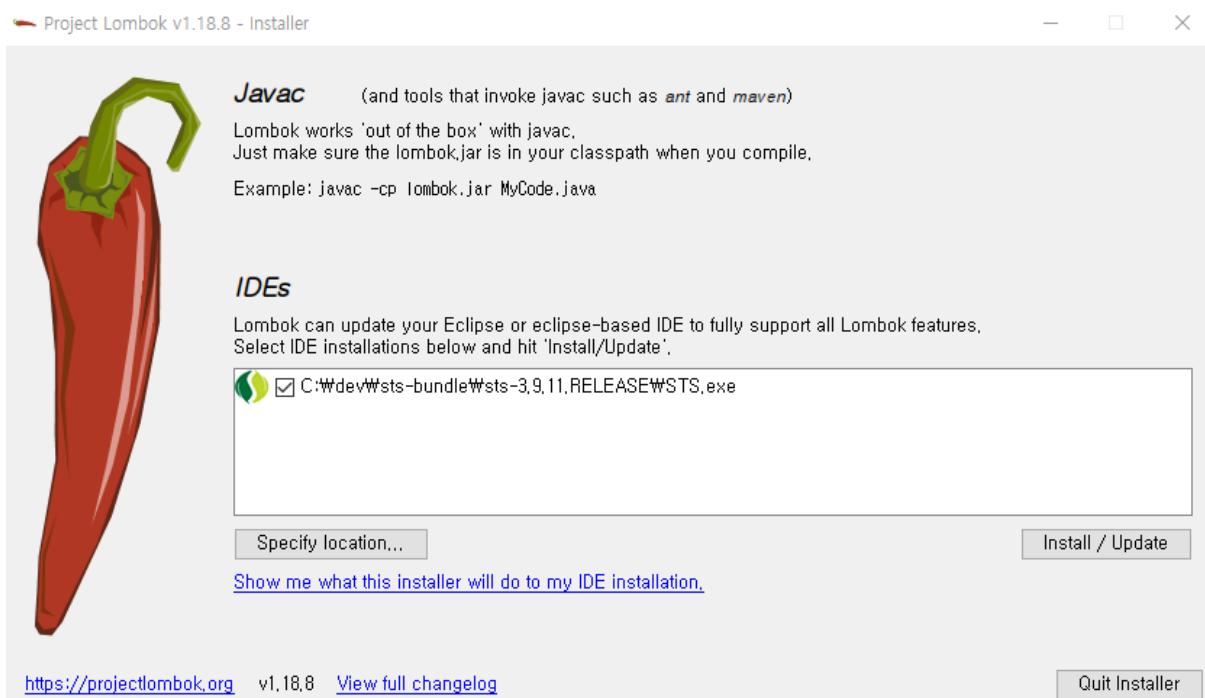
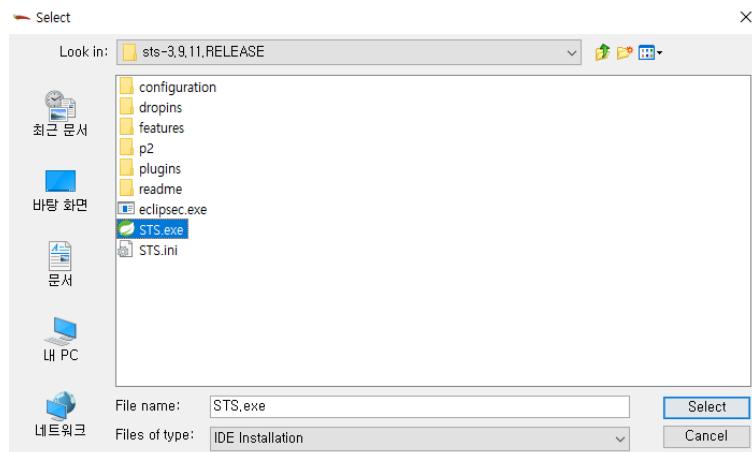
해당 명령어가 정상적으로 실행이 되면 아래와 같은 창이 뜨게 된다.



해당 화면에서 Specify location 버튼을 클릭하여 sts 실행 파일을 선택해준다.

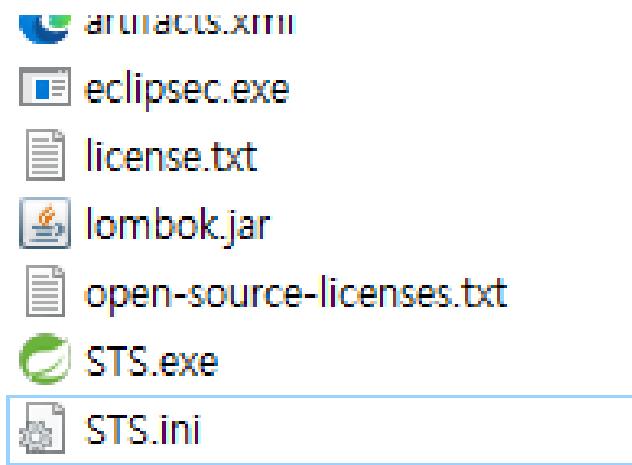


바탕화면에 바로가기 아이콘이 아닌 원래 경로에 있는 exe 파일을 선택한다.



그 뒤 Install/Update 버튼을 클릭 한 후

아래에 Quit Installer 버튼으로 설치를 마무리 한다.



그 뒤 sts 실행 파일이 있는 실제 경로에 가보면 lombok.jar 파일이 추가가 되어있고, STS.ini 파일을 열어보면 맨 아래에

javaagent:C:\dev\sts-bundle\sts-3.9.11.RELEASE\lombok.jar
문구가 추가 된 것을 확인할 수 있다.

그 뒤 생성하는 Legacy Project 의 pom.xml의 Dependency에 lombok을 추가해주어야 한다.

```
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.0</version>
    <scope>provided</scope>
</dependency>
```

!!! 디펜던시를 추가하고 난 뒤에는 메이븐 업데이트를 반드시 해줘야한다.

프로젝트 우클릭 → Maven → Update project

3. 추가 - 인코딩

요청과 응답에 따른 한글 깨짐 처리 인코딩 부분을 필터로 처리해도 되지만,
xml을 이용하여 설정해두도록 한다.

프로젝트 내의 web.xml 파일에 해당 코드를 추가한다.

```
<!-- 추가. 인코딩 -->
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>utf-8</param-value>
    </init-param>
    <init-param>
        <param-name>forceEncoding</param-name>
        <param-value>true</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

```
28< servlet-mapping>
29    <servlet-name>appServlet</servlet-name>
30        <url-pattern>/</url-pattern>
31    </servlet-mapping>
32
33
34    <!-- 추가. 인코딩 -->
35<filter>
36    <filter-name>encodingFilter</filter-name>
37    <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
38<init-param>
39    <param-name>encoding</param-name>
40    <param-value>utf-8</param-value>
41</init-param>
42<init-param>
43    <param-name>forceEncoding</param-name>
44    <param-value>true</param-value>
45</init-param>
46</filter>
47<filter-mapping>
48    <filter-name>encodingFilter</filter-name>
49    <url-pattern>/*</url-pattern>
50 </filter-mapping>
51
52
53
54 </web-app>
55
```