

# 1. sts 설치

## 1-1. sts 다운로드 및 설치

### (1) 다운로드 페이지로 이동 ➡ sts다운로드링크

The screenshot shows a browser window with the URL [github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3](https://github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3). The page content is as follows:

**Spring Tool Suite 3**  
Martin Lippert edited this page on Sep 20, 2021 · 11 revisions

**Important note**

The Spring Tool Suite 3 is the previous generation of the Spring tooling for the Eclipse IDE. The Spring Tool Suite distribution was based on the Eclipse JEE package and included the Spring IDE components, the tc Server integration for Eclipse, and various other pre-installed plugins for the Eclipse IDE.

With the release of the all-new Spring Tools 4, the Spring Tool Suite 3 got deprecated and is no longer under active development. Instead, users of the Spring Tool Suite 3 are highly encouraged to migrate to the all-new Spring Tools 4: <https://spring.io/tools>. Even though the new Spring Tools 4 are IDE-agnostic and available for various development tools, we continue to provide a ready-to-use Eclipse-based distribution with the necessary components pre-installed.

In case you can't migrate to the Spring Tools 4, please let us know and provide feedback about what you are missing within the new tools, what doesn't work for you, or what other issues you face. In such a case, please file an enhancement request at: <https://github.com/spring-projects/sts4/issues>. You can flag them, for example, as "sts3-feature-request" to make clear that this is something that you miss from the old tools.

For your convenience, we still update the Spring Tool Suite 3 to the latest Eclipse distributions and will probably continue to do that for a little while (based on community feedback and demand). You can find the download links below. But please keep in mind to migrate to the new Spring Tools 4, since we will not be able to update the Spring Tool Suite 3 distribution forever.

**Migrate to Spring Tools 4 for Eclipse**

On the right side of the page, there is a sidebar with the following sections:

- Pages (1)
  - Find a page...
- Spring Tool Suite 3
  - Important note
  - Migrate to Spring Tools 4 for Eclipse
  - Latest STS3 Downloads
    - Spring Tool Suite 3.9.18
      - full distribution on Eclipse 4.21
      - full distribution on Eclipse 4.20
      - full distribution on Eclipse 4.19
      - p2 repositories
    - Spring Tool Suite 3.9.17 (New and Noteworthy)
      - full distribution on Eclipse 4.20
  - Previous STS3 Versions
    - Spring Tool Suite 3.9.17 (New and Noteworthy)
      - full distribution on Eclipse 4.20

### (2) 아래와 같은 화면이 나올 때까지 스크롤합니다.

The screenshot shows a browser window with the URL [github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3](https://github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3). The page content is as follows:

**Latest STS3 Downloads**

**Spring Tool Suite 3.9.18**

full distribution on Eclipse 4.21      아래 링크를 클릭하여 다운로드합니다.

- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-win32-x86\\_64.zip](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-win32-x86_64.zip)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-macosx-cocoa-x86\\_64.dmg](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-macosx-cocoa-x86_64.dmg)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-linux-gtk-x86\\_64.tar.gz](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.21/spring-tool-suite-3.9.18.RELEASE-e4.21.0-linux-gtk-x86_64.tar.gz)

full distribution on Eclipse 4.20

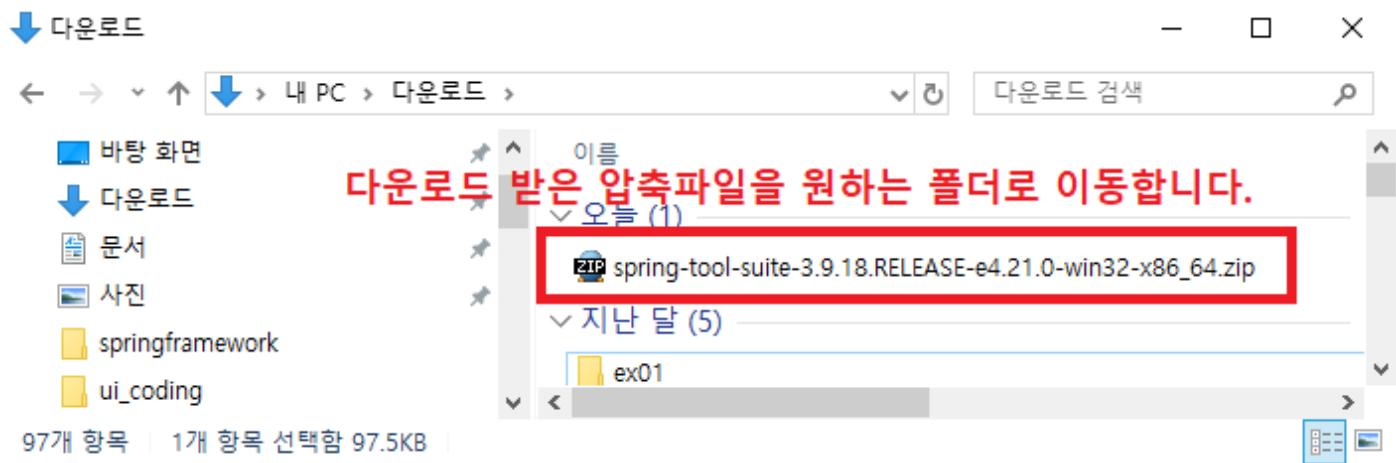
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-win32-x86\\_64.zip](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-win32-x86_64.zip)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-macosx-cocoa-x86\\_64.dmg](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-macosx-cocoa-x86_64.dmg)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-linux-gtk-x86\\_64.tar.gz](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.20/spring-tool-suite-3.9.18.RELEASE-e4.20.0-linux-gtk-x86_64.tar.gz)

full distribution on Eclipse 4.19

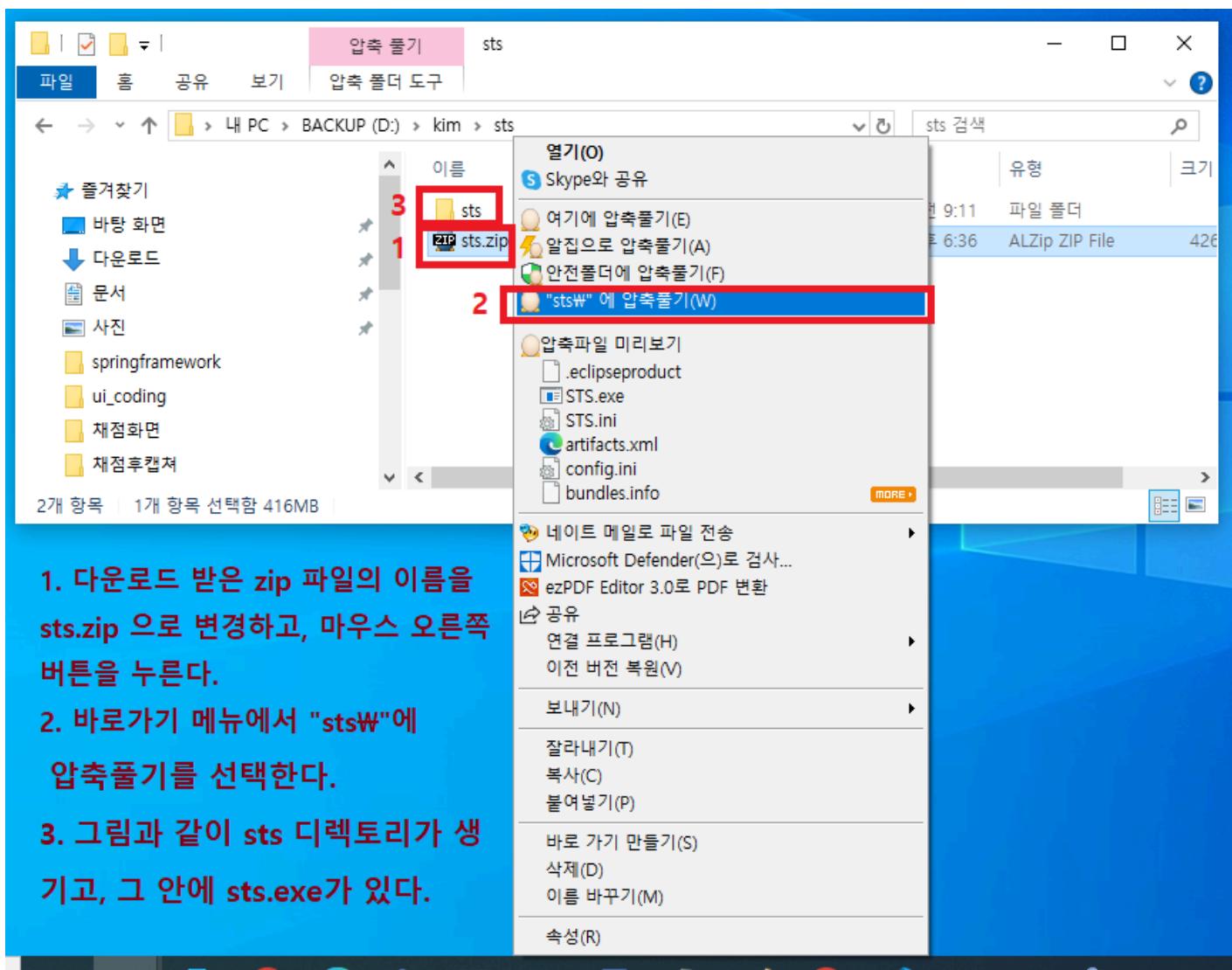
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-win32-x86\\_64.zip](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-win32-x86_64.zip)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-macosx-cocoa-x86\\_64.dmg](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-macosx-cocoa-x86_64.dmg)
- [https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-linux-gtk-x86\\_64.tar.gz](https://download.springsource.com/release/STS/3.9.18.RELEASE/dist/e4.19/spring-tool-suite-3.9.18.RELEASE-e4.19.0-linux-gtk-x86_64.tar.gz)

On the right side of the page, there is a sidebar with the following sections:

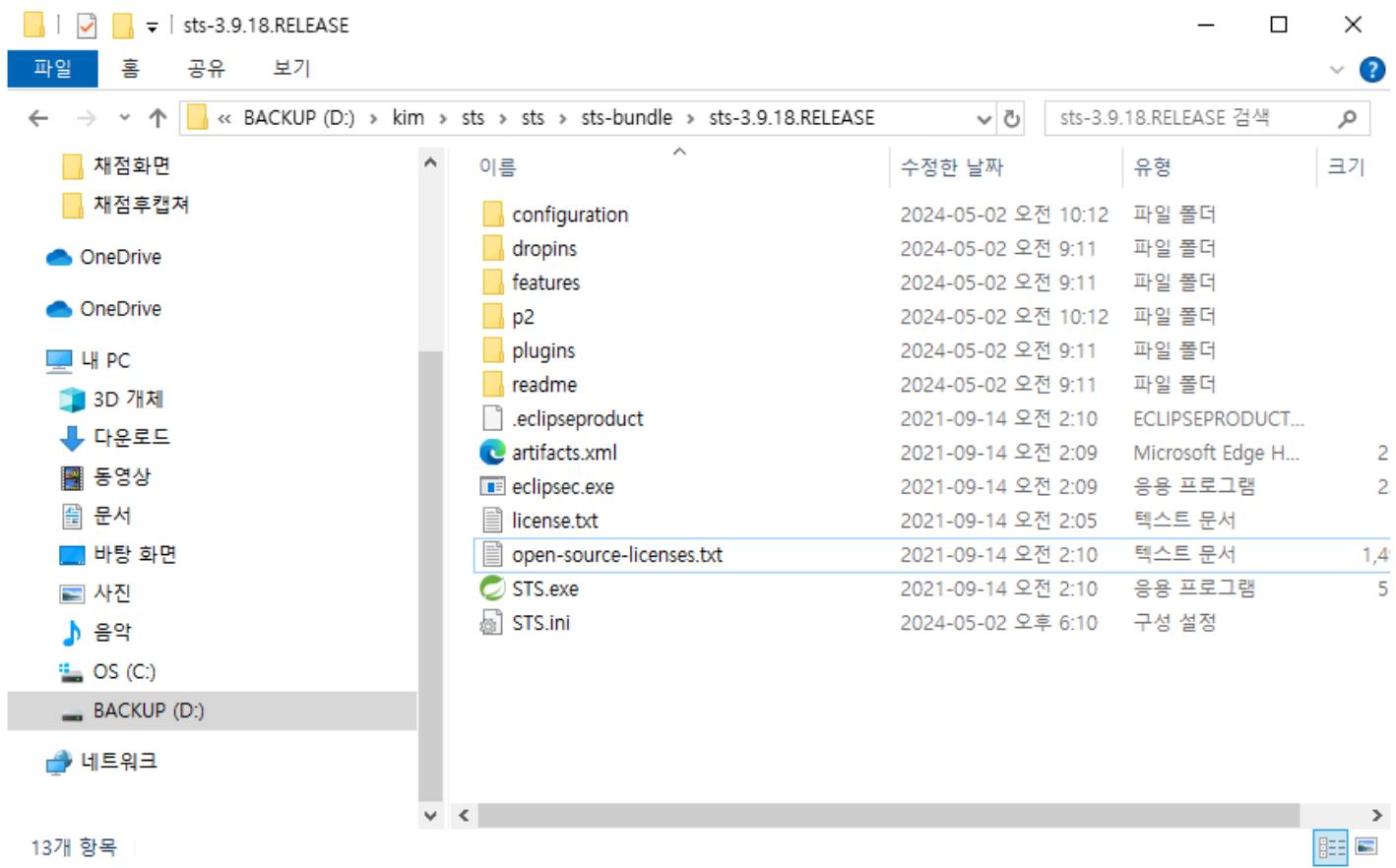
- p2 repositories
  - Spring Tool Suite 3.9.16 (New and Noteworthy)
    - full distribution on Eclipse 4.19
    - full distribution on Eclipse 4.18
    - p2 repositories
  - Spring Tool Suite 3.9.15 (New and Noteworthy)
    - full distribution on Eclipse 4.18
    - full distribution on Eclipse 4.17
    - full distribution on Eclipse 4.16
    - p2 repositories
  - Spring Tool Suite 3.9.14 (New and Noteworthy)
    - full distribution on Eclipse 4.17
    - full distribution on Eclipse 4.16
    - full distribution on Eclipse 4.15
    - p2 repositories
  - Spring Tool Suite 3.9.13 (New and Noteworthy)
    - full distribution on Eclipse 4.16
    - full distribution on Eclipse 4.15
    - full distribution on Eclipse 4.14



(3) 압축 파일을 원하는 폴더로 이동후에 이름을 sts.zip으로 변경하고, 아래 그림과 같이 진행합니다.



(4) 압축풀기가 끝나면 아래 그림과 같이 \sts\sts-bundle\sts-3.9.18RELEASE 폴더로 이동합니다.



## 1-2. Lombok 설치

※ Lombok을 활용하면, DTO 클래스에 getter/setter/toString/Constructor 의 정의가 필요 없게 되어 프로젝트 진행 시 개발 시간을 단축시킬 수 있습니다.

(1) DTO 생성을 위한 Lombok.jar를 설치하기 위해 mvnrepository 페이지로 이동합니다.

mvnrepository.com 해당 URL로 이동합니다.

## MVN REPOSITORY

Indexed Artifacts (35.4M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries
- HTTP Clients
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks
- I/O Utilities
- Defect Detection Metadata
- Code Generators
- Configuration Libraries

### What's New in Maven

**PrimeFaces**  
org.primefaces > primefaces > 13.0.9  
Last Release on May 1, 2024

**Contract Module**  
com.ritense.valtimo > contract > 11.3.1.RELEASE  
The contract module is a container for interfaces, or contracts between implementations and implementation consumers  
Last Release on May 1, 2024

**Core Module**  
com.ritense.valtimo > core > 11.3.1.RELEASE  
The core module is a wrapper around the BPMN engine, allowing you to launch and control business process models  
Last Release on May 1, 2024

**Document Module**  
com.ritense.valtimo > document > 11.3.1.RELEASE  
The document module allows you to define data schema for a business process  
Last Release on May 1, 2024

**CDK8s**  
org.cdk8s > cdk8s > 2.68.65  
This is the core library of Cloud Development Kit (CDK) for Kubernetes (cdk8s). cdk8s apps synthesize into standard Kubernetes manifests which can be applied to any Kubernetes cluster.  
Last Release on May 1, 2024

(2) Lombok을 검색하여 해당 카테고리를 클릭하고, 상세 페이지로 이동합니다.

mvnrepository.com/search?q=lombok

## MVN REPOSITORY

lombok 1. lombok 입력 2. Search 버튼 클릭

Found 150 results

Sort: relevance | popular | newest

- Project Lombok  
org.projectlombok > lombok  
23,341 usages MIT  
1. Project Lombok  
Last Release on Mar 20, 2024
- Lombok Mapstruct Binding  
org.projectlombok > lombok-mapstruct-binding  
33 usages MIT  
2. Lombok Mapstruct Binding  
Binding for Lombok and Mapstruct, to allow them to cooperate.  
Last Release on Dec 4, 2020
- Lombok Maven Plugin  
org.projectlombok > lombok-maven-plugin  
25 usages MIT  
3. Lombok Maven Plugin  
Maven Plugin for Project Lombok  
Last Release on Apr 2, 2021
- Collection of Lombok Related Gradle Plugins  
io.freefair.gradle > lombok-plugin  
13 usages MIT  
4. Collection of Lombok Related Gradle Plugins  
Collection of Lombok related Gradle plugins  
Last Release on Feb 16, 2024
- Maven Plugin For Project Lombok  
org.projectlombok > lombok-maven  
12 usages MIT  
5. Maven Plugin For Project Lombok

ads.travelaudience.com의 응답을 기다리는 중...

(3) Lombok의 Archive 목록에서 원하는 버전을 클릭하여 선택합니다.

mvnrepository.com/artifact/org.projectlombok/lombok

**MVN REPOSITORY**

Indexed Artifacts (35.4M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries
- HTTP Clients
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks
- I/O Utilities
- Defect Detection Metadata
- Code Generators
- Configuration Libraries
- OSGI Utilities
- Concurrency Libraries
- Reflection Libraries
- Android Platform
- JDBC Drivers
- Date and Time Utilities

Project Lombok

Lombok is a Java library that provides annotations to simplify Java development by automating the generation of boilerplate code. Key features include automatic generation of getters, setters, equals, hashCode, and toString methods, as well as a facility for automatic resource management. It aims to reduce the amount of manual coding, thereby streamlining the codebase and reducing potential for errors.

License	MIT																																																																																																						
Categories	Code Generators																																																																																																						
Tags	lombok, codegen, code, generator																																																																																																						
Ranking	#13 in MvnRepository (See Top Artifacts) #1 in Code Generators																																																																																																						
Used By	23,341 artifacts																																																																																																						
Central (52)	Codemc Public (1)	Lombok (7)	ICM (5)																																																																																																				
<table border="1"> <thead> <tr> <th>Version</th> <th>Vulnerabilities</th> <th>Repository</th> <th>Usages</th> <th>Date</th> </tr> </thead> <tbody> <tr><td>1.18.32</td><td></td><td>Central</td><td>521</td><td>Mar 20, 2024</td></tr> <tr><td>1.18.30</td><td></td><td>Central</td><td>2,893</td><td>Sep 20, 2023</td></tr> <tr><td>1.18.28</td><td></td><td>Central</td><td>1,692</td><td>May 25, 2023</td></tr> <tr><td>1.18.26</td><td></td><td>Central</td><td>1,962</td><td>Feb 03, 2023</td></tr> <tr><td>1.18.24</td><td></td><td>Central</td><td>4,254</td><td>Apr 18, 2022</td></tr> <tr><td>1.18.22</td><td>클릭</td><td>Central</td><td>3,334</td><td>Oct 07, 2021</td></tr> <tr><td>1.18.20</td><td></td><td>Central</td><td>2,954</td><td>Apr 02, 2021</td></tr> <tr><td>1.18.18</td><td></td><td>Central</td><td>873</td><td>Jan 28, 2021</td></tr> <tr><td>1.18.x</td><td>1.18.16</td><td>Central</td><td>1,970</td><td>Oct 15, 2020</td></tr> <tr><td></td><td>1.18.14</td><td>Central</td><td>213</td><td>Oct 08, 2020</td></tr> <tr><td></td><td>1.18.12</td><td>Central</td><td>3,215</td><td>Feb 07, 2020</td></tr> <tr><td></td><td>1.18.10</td><td>Central</td><td>1,804</td><td>Sep 11, 2019</td></tr> <tr><td></td><td>1.18.8</td><td>Central</td><td>1,757</td><td>May 07, 2019</td></tr> <tr><td></td><td>1.18.6</td><td>Central</td><td>1,223</td><td>Feb 12, 2019</td></tr> <tr><td></td><td>1.18.4</td><td>Central</td><td>1,163</td><td>Oct 30, 2018</td></tr> <tr><td></td><td>1.18.2</td><td>Central</td><td>977</td><td>Jul 25, 2018</td></tr> <tr><td></td><td>1.18.0</td><td>Central</td><td>433</td><td>Jun 04, 2018</td></tr> <tr><td></td><td>1.16.22</td><td>Central</td><td>518</td><td>May 28, 2018</td></tr> <tr><td></td><td>1.16.20</td><td>Central</td><td>1,149</td><td>Jan 08, 2018</td></tr> </tbody> </table>				Version	Vulnerabilities	Repository	Usages	Date	1.18.32		Central	521	Mar 20, 2024	1.18.30		Central	2,893	Sep 20, 2023	1.18.28		Central	1,692	May 25, 2023	1.18.26		Central	1,962	Feb 03, 2023	1.18.24		Central	4,254	Apr 18, 2022	1.18.22	클릭	Central	3,334	Oct 07, 2021	1.18.20		Central	2,954	Apr 02, 2021	1.18.18		Central	873	Jan 28, 2021	1.18.x	1.18.16	Central	1,970	Oct 15, 2020		1.18.14	Central	213	Oct 08, 2020		1.18.12	Central	3,215	Feb 07, 2020		1.18.10	Central	1,804	Sep 11, 2019		1.18.8	Central	1,757	May 07, 2019		1.18.6	Central	1,223	Feb 12, 2019		1.18.4	Central	1,163	Oct 30, 2018		1.18.2	Central	977	Jul 25, 2018		1.18.0	Central	433	Jun 04, 2018		1.16.22	Central	518	May 28, 2018		1.16.20	Central	1,149	Jan 08, 2018
Version	Vulnerabilities	Repository	Usages	Date																																																																																																			
1.18.32		Central	521	Mar 20, 2024																																																																																																			
1.18.30		Central	2,893	Sep 20, 2023																																																																																																			
1.18.28		Central	1,692	May 25, 2023																																																																																																			
1.18.26		Central	1,962	Feb 03, 2023																																																																																																			
1.18.24		Central	4,254	Apr 18, 2022																																																																																																			
1.18.22	클릭	Central	3,334	Oct 07, 2021																																																																																																			
1.18.20		Central	2,954	Apr 02, 2021																																																																																																			
1.18.18		Central	873	Jan 28, 2021																																																																																																			
1.18.x	1.18.16	Central	1,970	Oct 15, 2020																																																																																																			
	1.18.14	Central	213	Oct 08, 2020																																																																																																			
	1.18.12	Central	3,215	Feb 07, 2020																																																																																																			
	1.18.10	Central	1,804	Sep 11, 2019																																																																																																			
	1.18.8	Central	1,757	May 07, 2019																																																																																																			
	1.18.6	Central	1,223	Feb 12, 2019																																																																																																			
	1.18.4	Central	1,163	Oct 30, 2018																																																																																																			
	1.18.2	Central	977	Jul 25, 2018																																																																																																			
	1.18.0	Central	433	Jun 04, 2018																																																																																																			
	1.16.22	Central	518	May 28, 2018																																																																																																			
	1.16.20	Central	1,149	Jan 08, 2018																																																																																																			

#### (4) jar 파일을 다운로드합니다.

mvnrepository.com/artifact/org.projectlombok/lombok/1.18.22

**MVN REPOSITORY**

Indexed Artifacts (35.4M)

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Language Runtime
- Core Utilities
- Mocking
- Web Assets
- Annotation Libraries
- HTTP Clients
- Logging Bridges
- Dependency Injection
- XML Processing
- Web Frameworks
- I/O Utilities
- Defect Detection Metadata
- Code Generators
- Configuration Libraries
- OSGI Utilities
- Concurrency Libraries
- Reflection Libraries
- Android Platform
- JDBC Drivers
- Date and Time Utilities

Project Lombok » 1.18.22

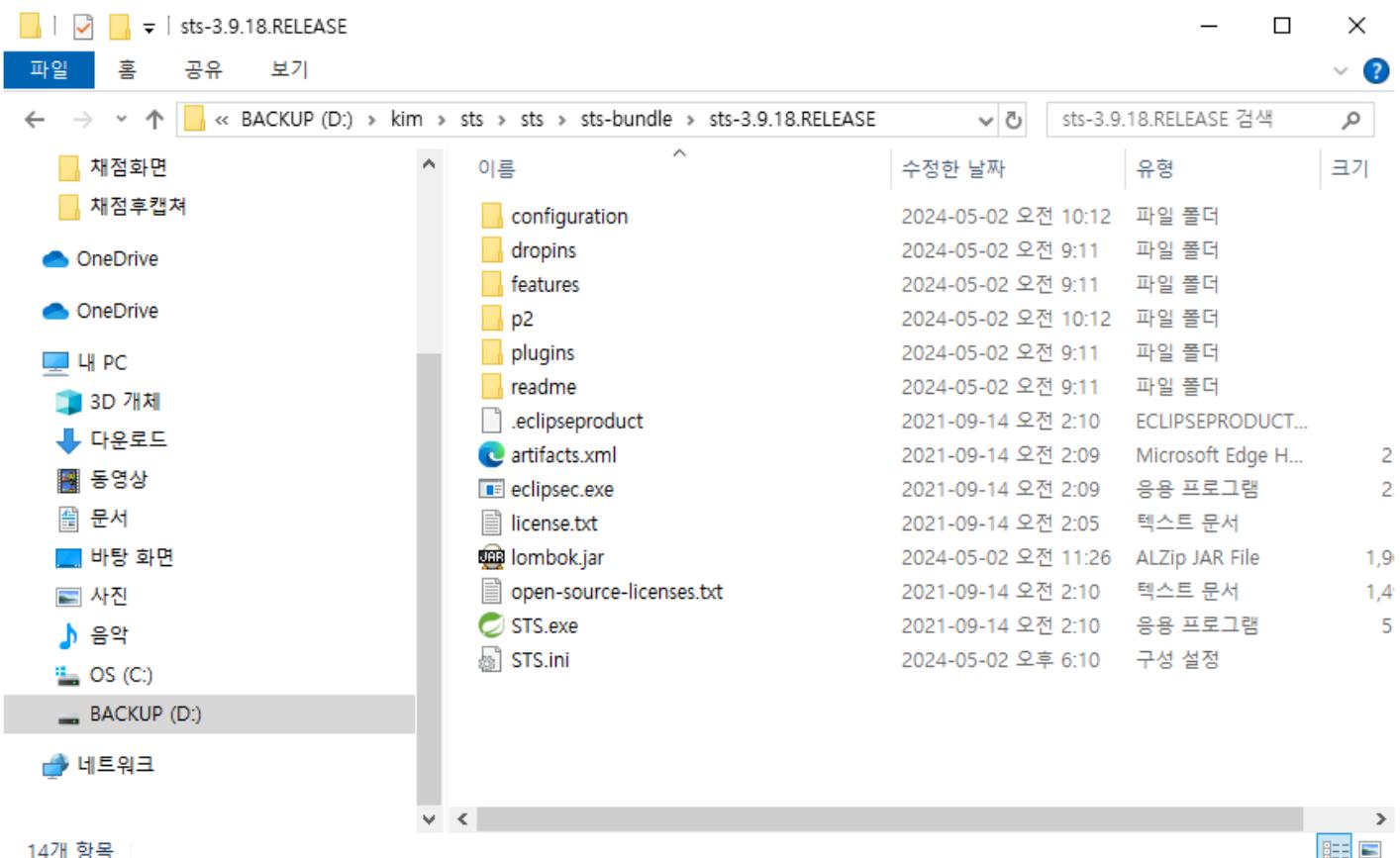
Lombok is a Java library that provides annotations to simplify Java development by automating the generation of boilerplate code. Key features include automatic generation of getters, setters, equals, hashCode, and toString methods, as well as a facility for automatic resource management. It aims to reduce the amount of manual coding, thereby streamlining the codebase and reducing potential for errors.

License	MIT								
Categories	Code Generators								
Tags	lombok, codegen, code, generator								
HomePage	<a href="https://projectlombok.org">https://projectlombok.org</a>								
Date	Oct 07, 2021								
Files	pom (1 KB) <a href="#">jar (1.9 MB)</a> <a href="#">View All</a>								
Repositories	Central   Mulesoft								
Ranking	#13 in MvnRepository (See Top Artifacts) #1 in Code Generators								
Used By	23,341 artifacts								
<p>Note: There is a new version for this artifact</p> <p>New Version 1.18.32</p>									
<p>Maven   Gradle   Gradle (Short)   Gradle (Kotlin)   SBT   Ivy   Grape   Leiningen   Buildr</p> <pre>&lt;!-- https://mvnrepository.com/artifact/org.projectlombok/lombok --&gt; &lt;dependency&gt;     &lt;groupId&gt;org.projectlombok&lt;/groupId&gt;     &lt;artifactId&gt;lombok&lt;/artifactId&gt;     &lt;version&gt;1.18.22&lt;/version&gt;     &lt;scope&gt;provided&lt;/scope&gt; &lt;/dependency&gt;</pre> <p><input checked="" type="checkbox"/> Include comment with link to declaration</p>									
<p>Compile Dependencies (0)</p> <table border="1"> <thead> <tr> <th>Category/License</th> <th>Group / Artifact</th> <th>Version</th> <th>Updates</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>		Category/License	Group / Artifact	Version	Updates				
Category/License	Group / Artifact	Version	Updates						

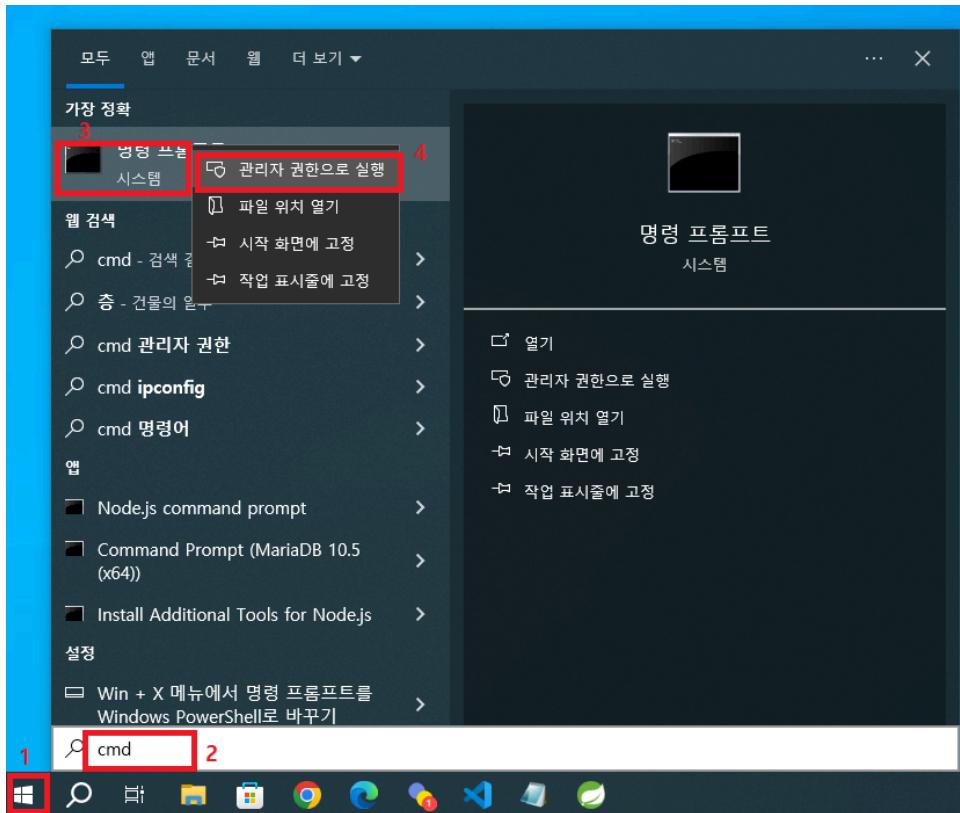
#### (5) download 폴더로 이동하여 다운로드된 lombok-1.18.22.jar파일을 찾아 파일의 이름을 lombok.jar로 변경합니다.



(6) lombok.jar 파일을 잘라내기하여 sts가 설치된 폴더로 붙여넣기 합니다.



(7) cmd(명령 프롬프트)를 관리자모드로 실행합니다.



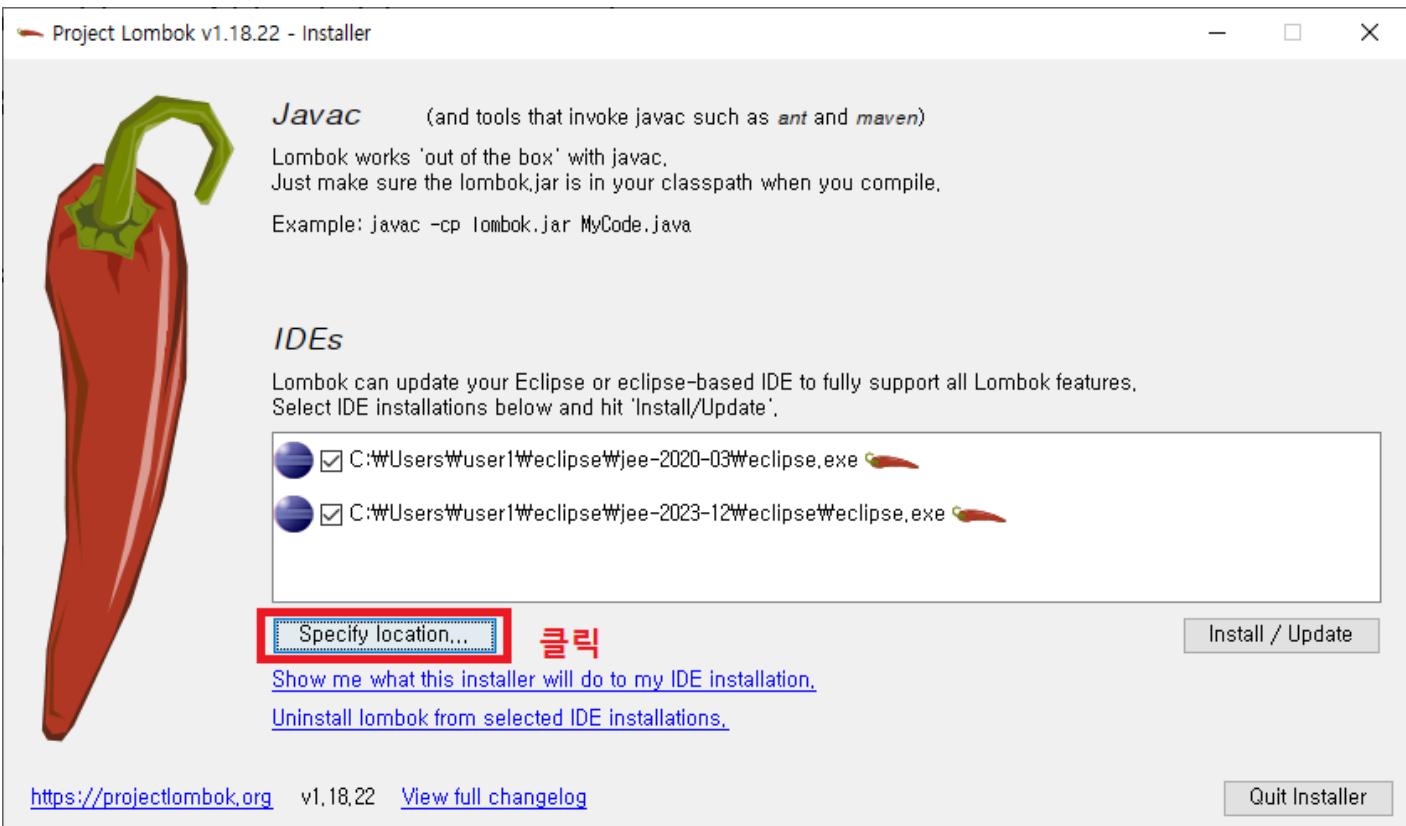
1. 원도우 버튼을 누릅니다.
2. cmd 를 입력합니다.
3. 메뉴에서 명령 프롬프트 에서 마우스 오른쪽 버튼을 누릅니다.
4. 바로가기 메뉴에서 관리자 권한으로 실행을 합니다.

(8) 해당 디렉토리로 이동 후에 java의 jar 명령으로 lombok.jar의 압축을 풀면서 실행을 시킵니다.

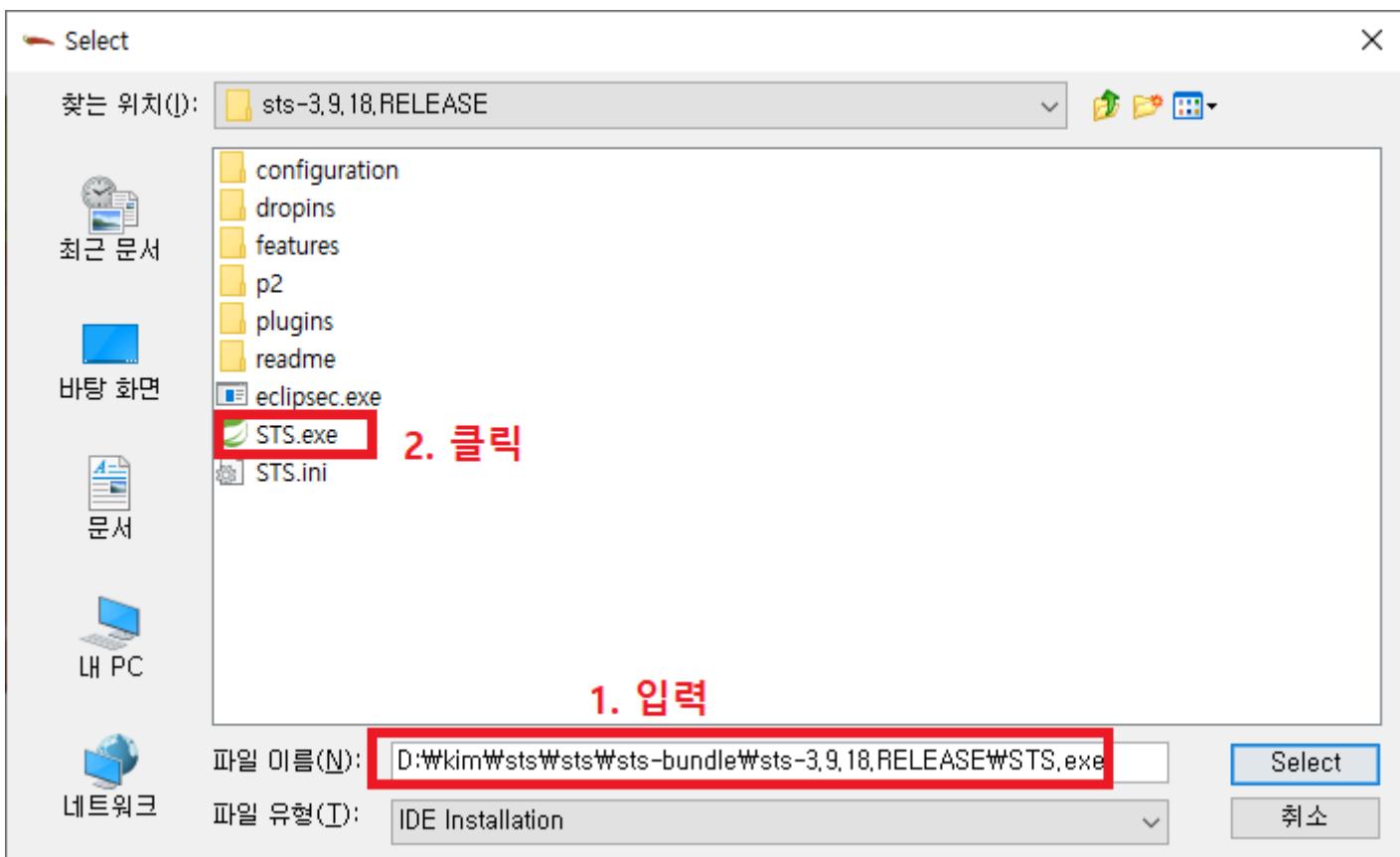
```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>d:    입력 후 엔터
D:>cd \kim\sts\sts-bundle\sts-3.9.18.RELEASE    입력 후 엔터
D:\kim\sts\sts-bundle\sts-3.9.18.RELEASE>java -jar lombok.jar    입력 후 엔터
```

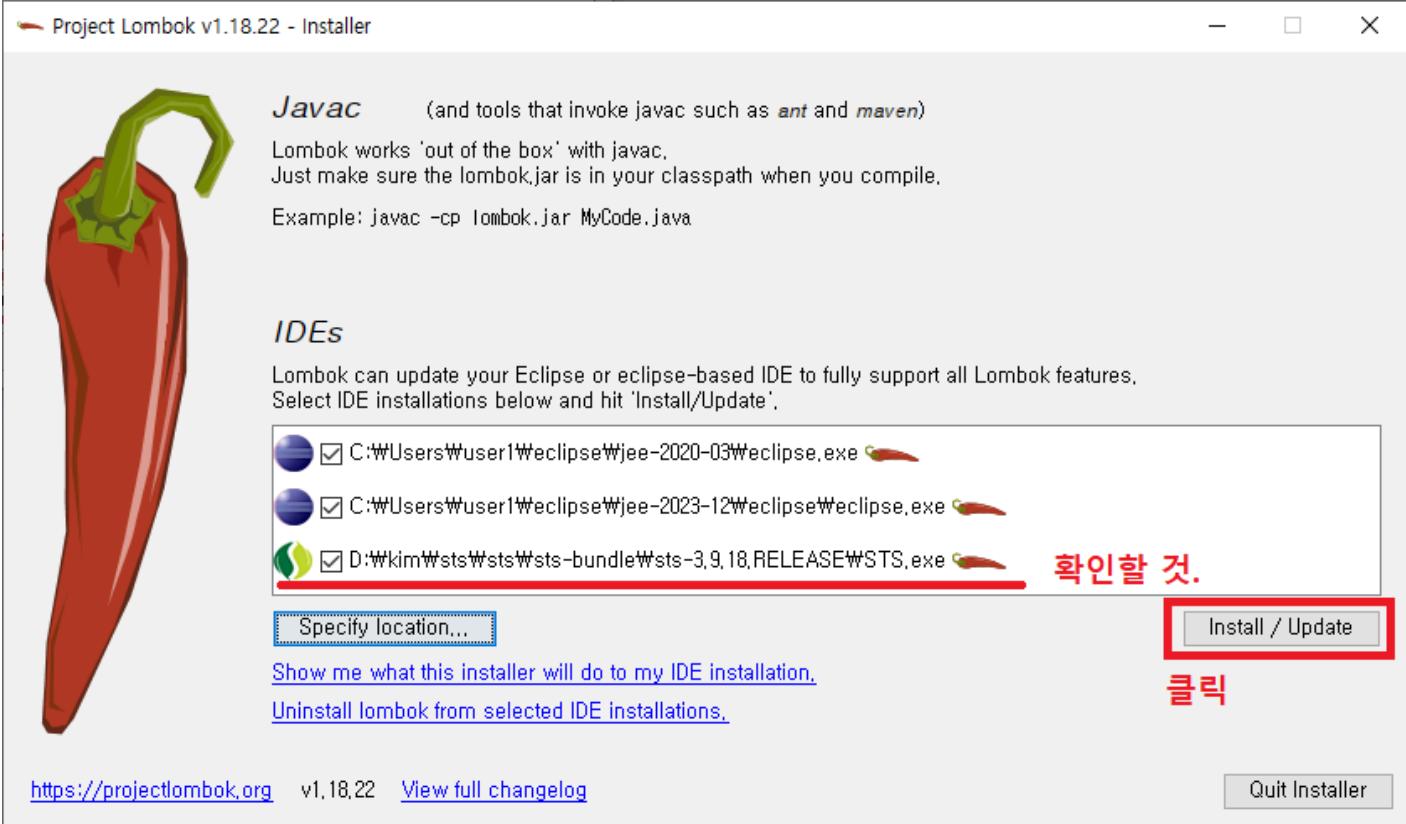
(9) lombok 설치 화면에서 [Specify location] 버튼을 클릭합니다.



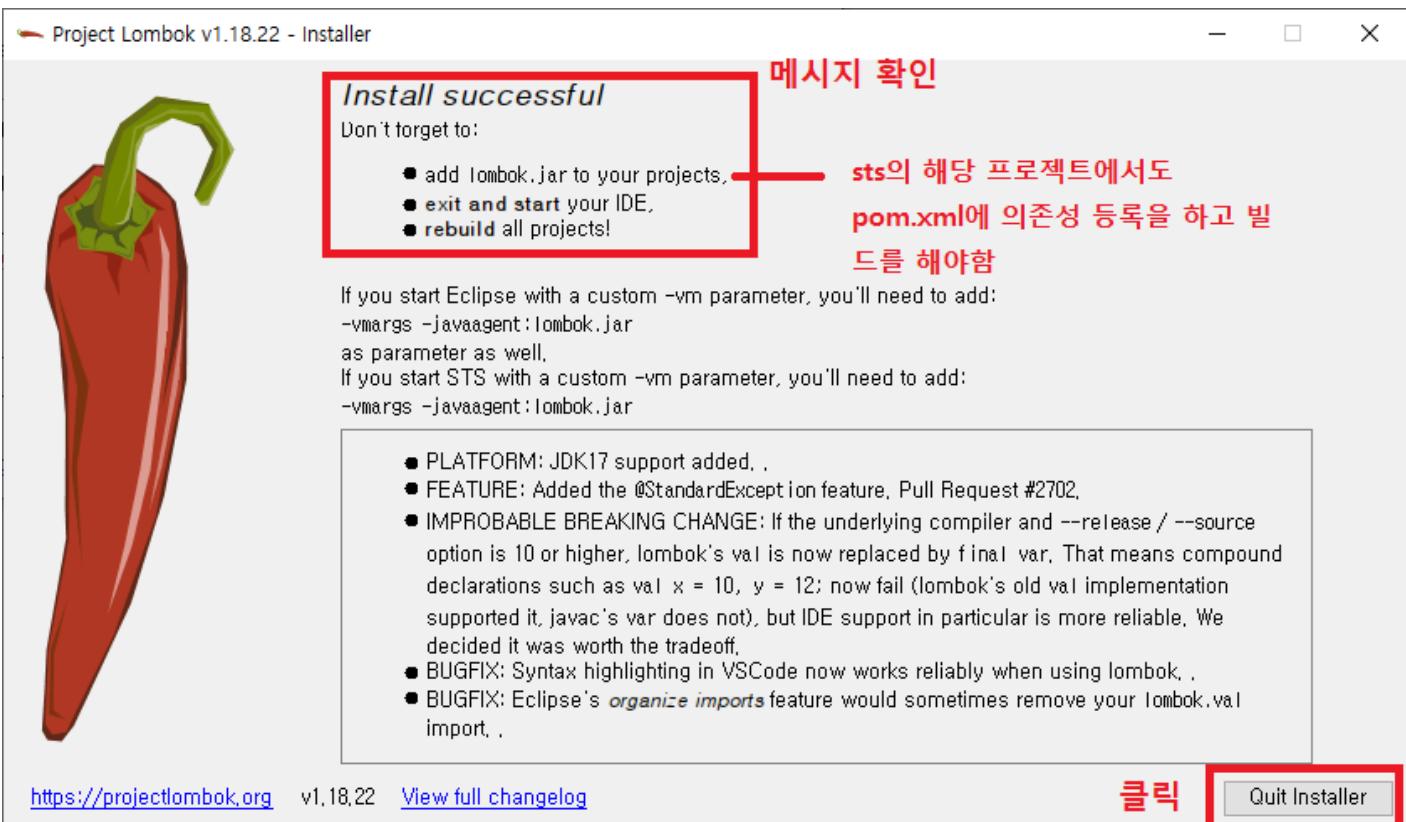
(10) sts가 설치된 디렉토리 및 STS.exe 파일의 경로를 지정합니다.



(11) 경로지정이 끝나면 설치를 진행합니다.

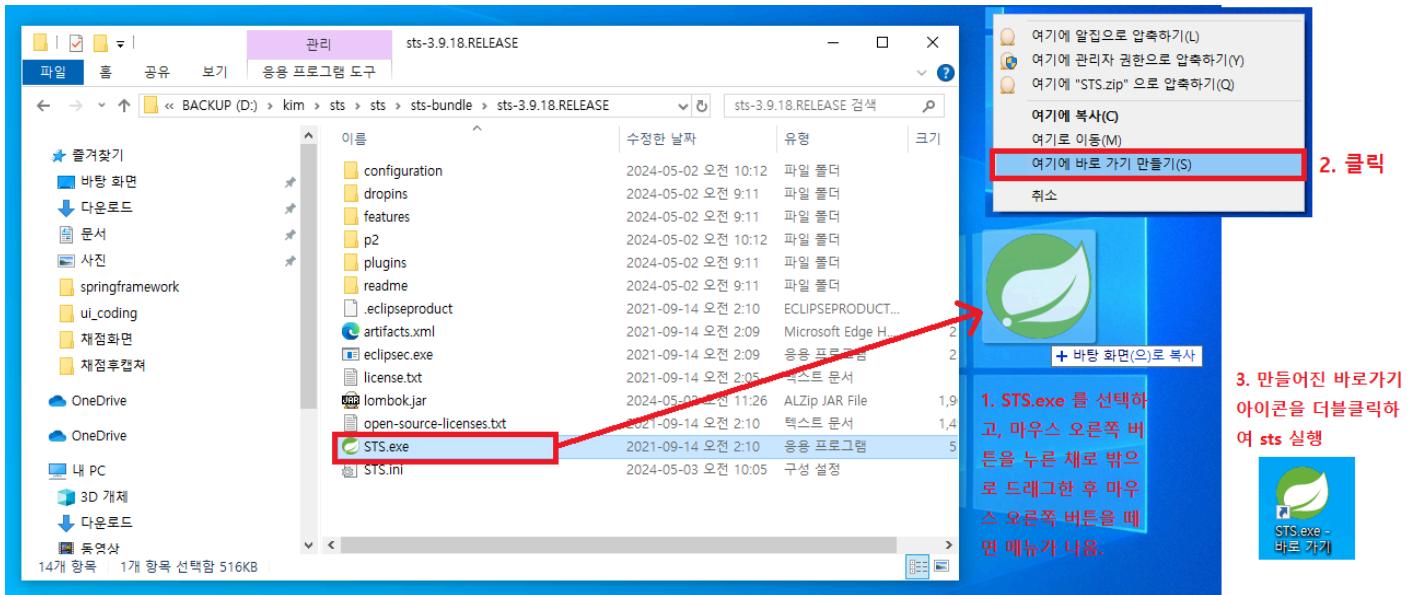


(12) 설치가 모두 마치면, Lombok 설치를 종료합니다.

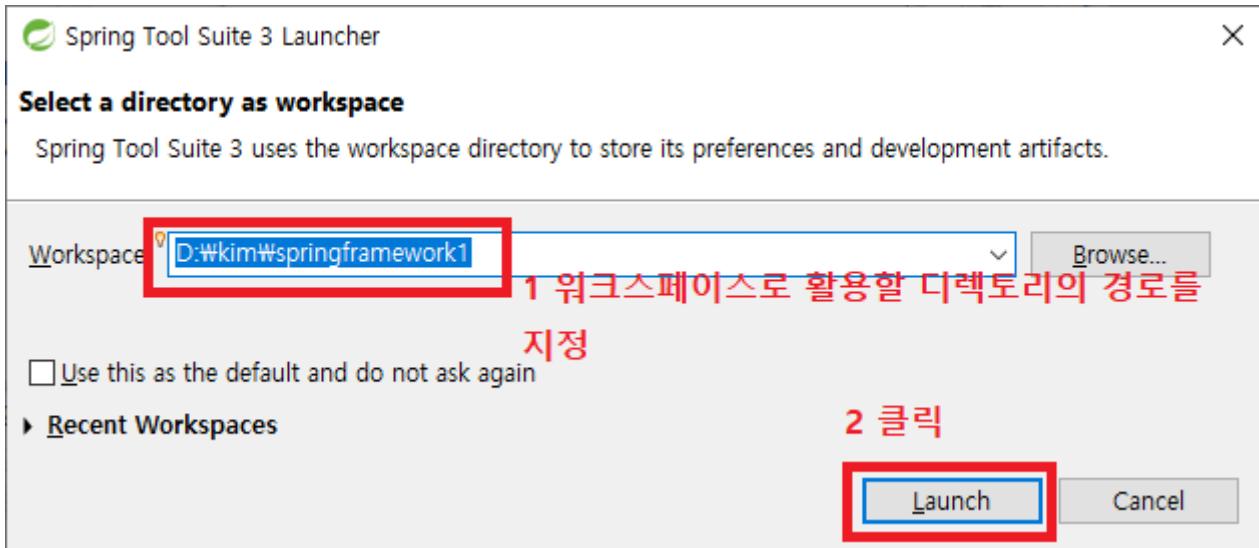


1-3. sts 설정

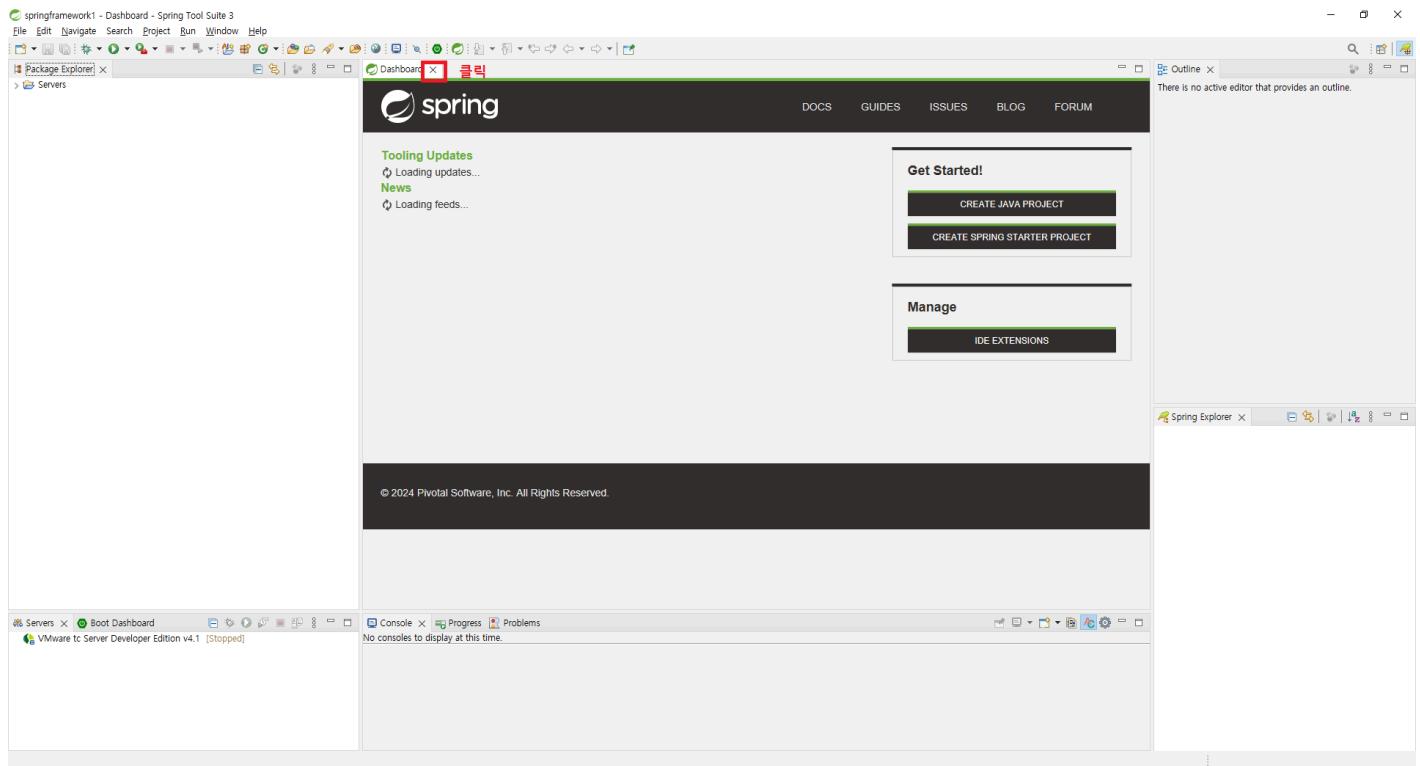
## (1) sts의 바로가기 아이콘을 바탕화면에 만들어 실행합니다.



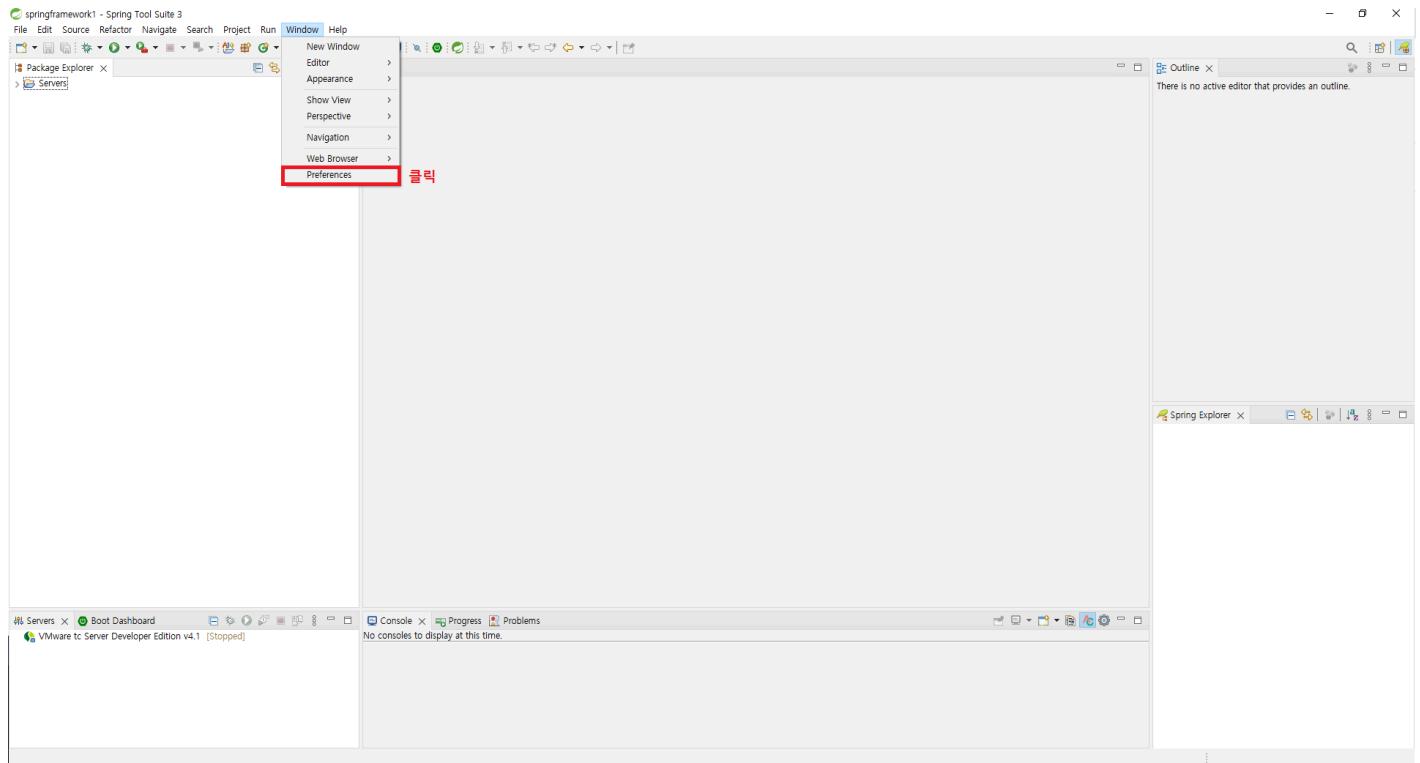
## (2) 프로젝트의 디렉토리인 워크스페이스 디렉토리를 지정합니다.



## (3) Dash 보드 패널을 닫습니다.

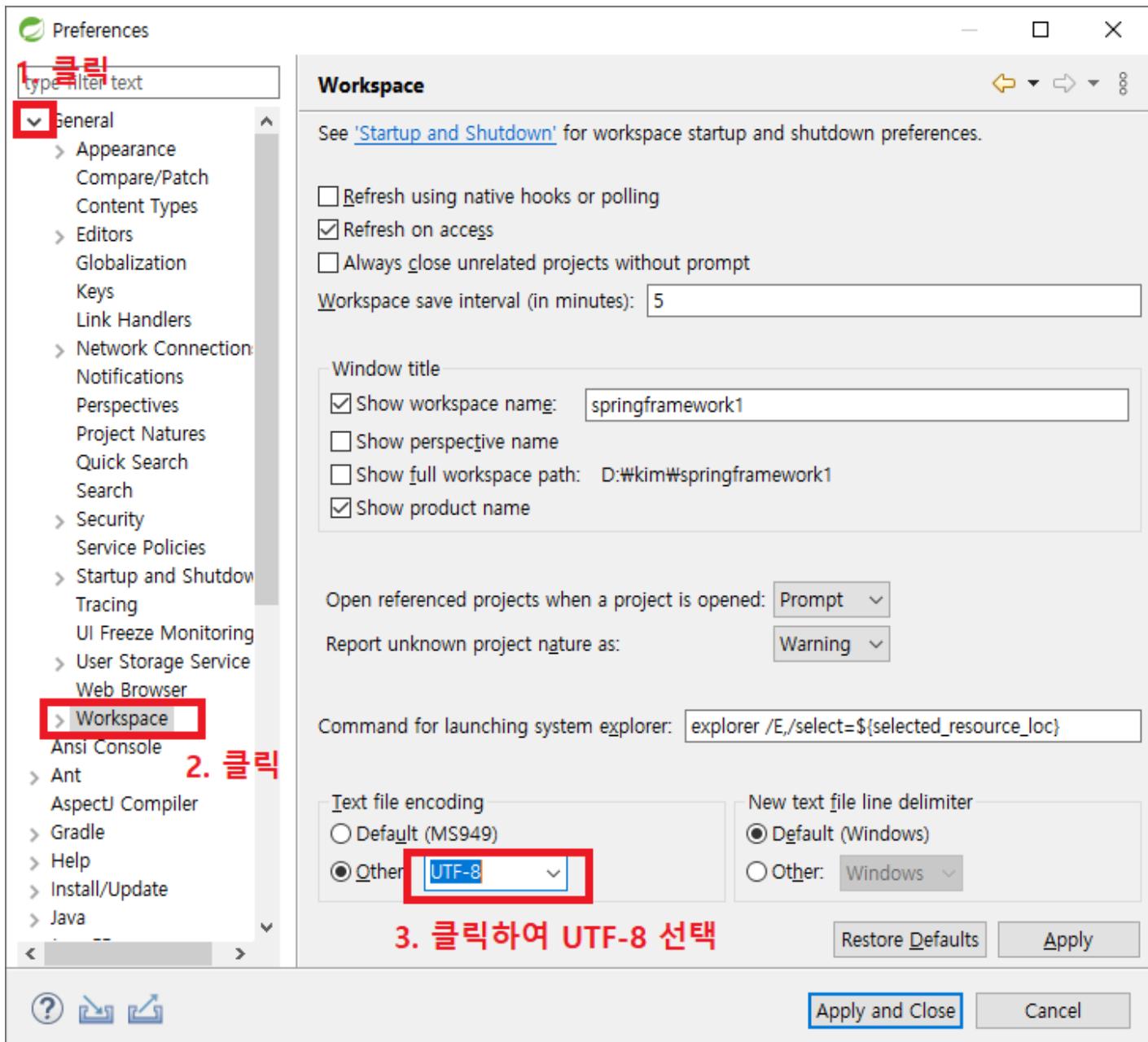


(4) [Windows]-[Properties] 메뉴를 통하여 환경 설정 화면으로 이동합니다.

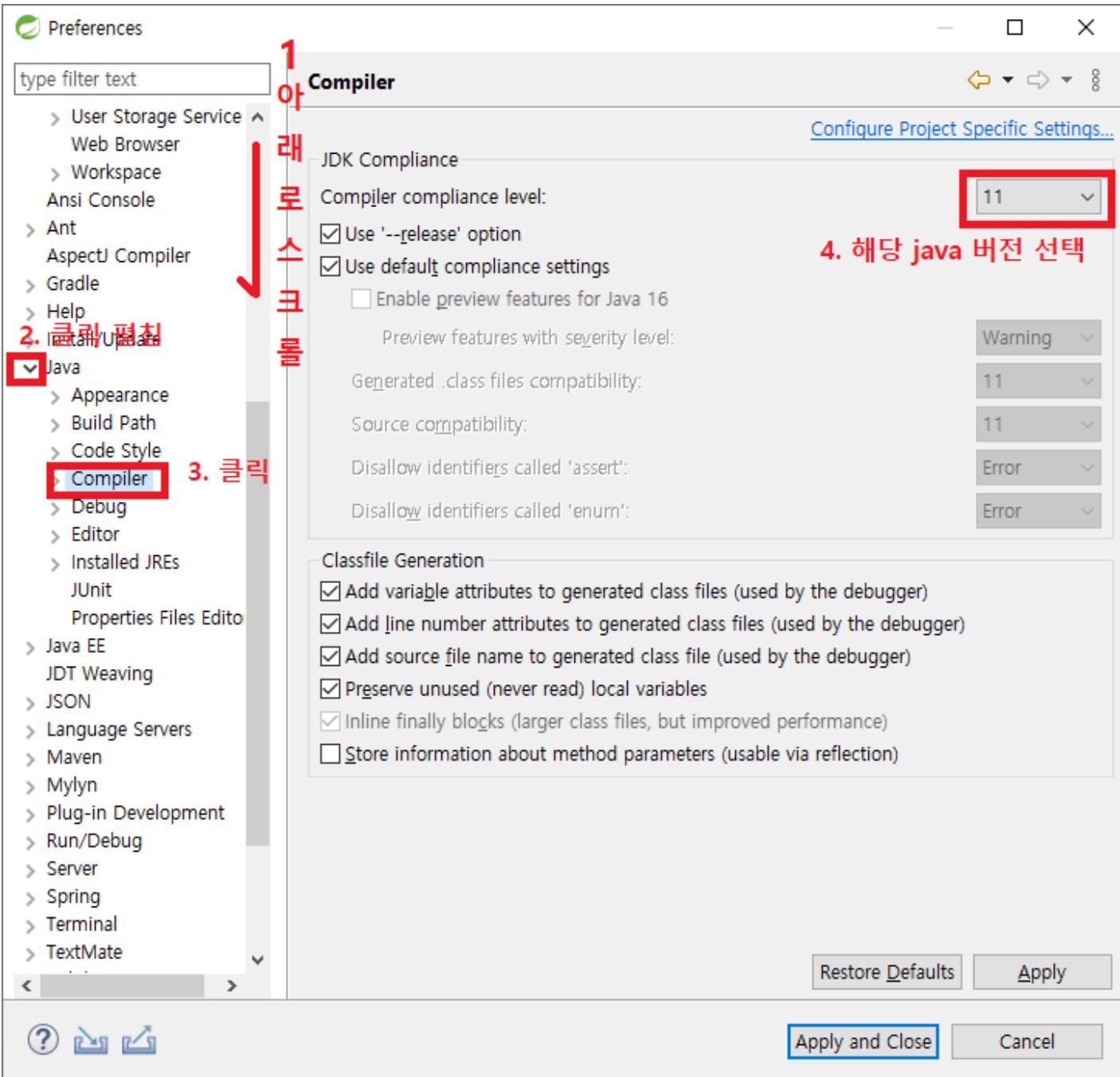


※ 워크스페이스를 변경시에는 환경 설정을 다시 해야합니다.

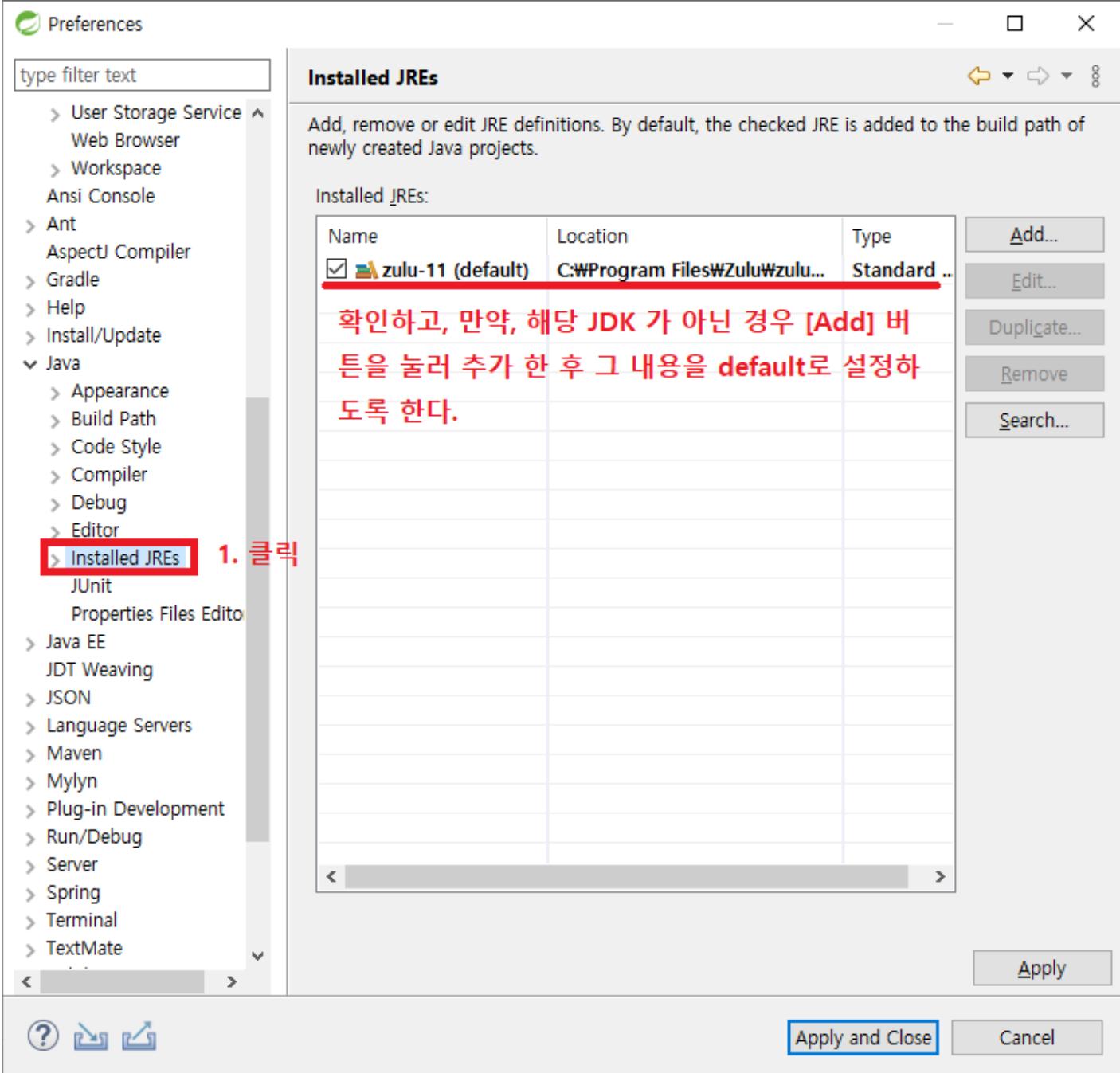
(5) 워크스페이스의 인코딩을 설정합니다.



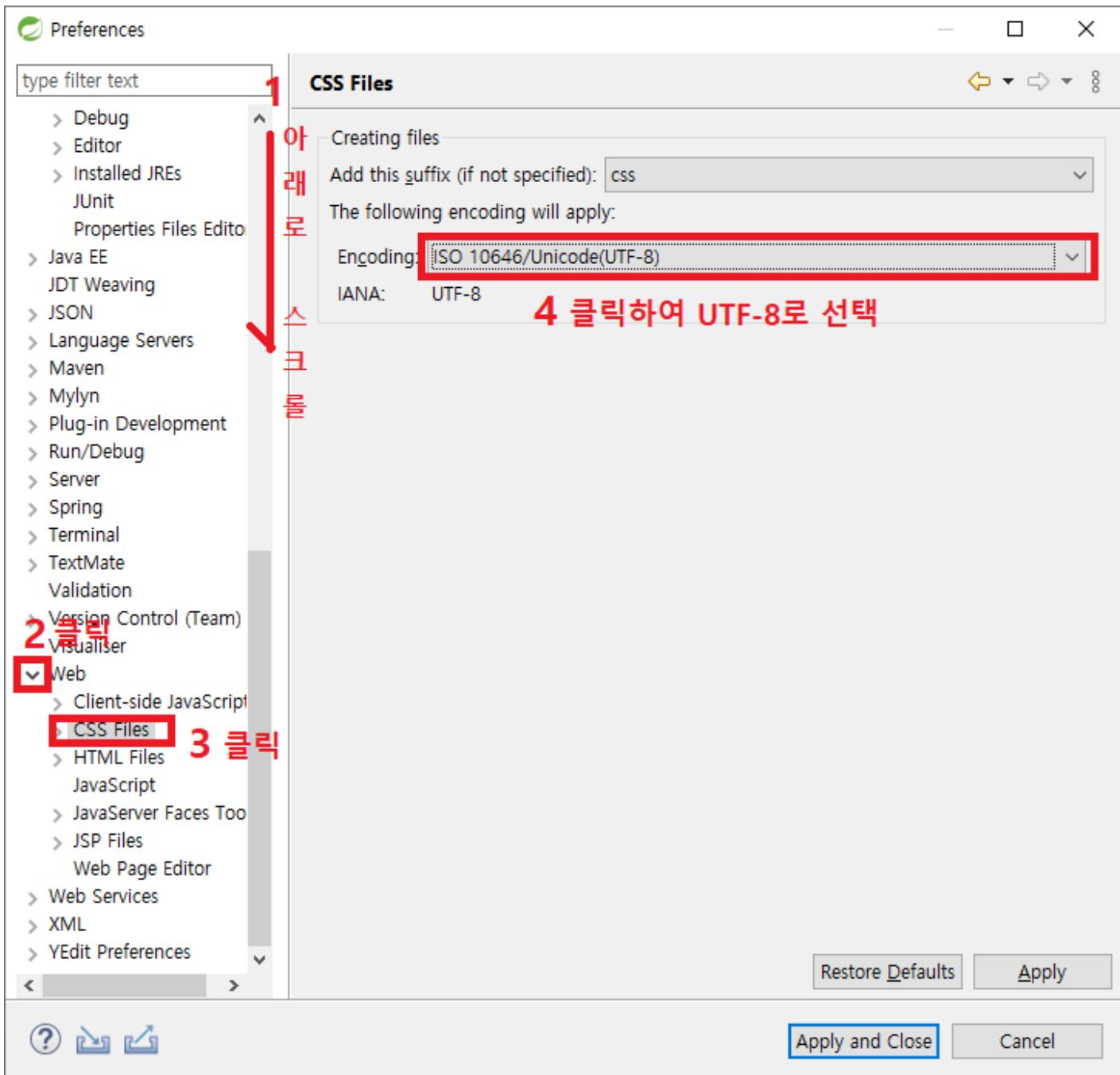
(6) 자바 컴파일러를 설정합니다.



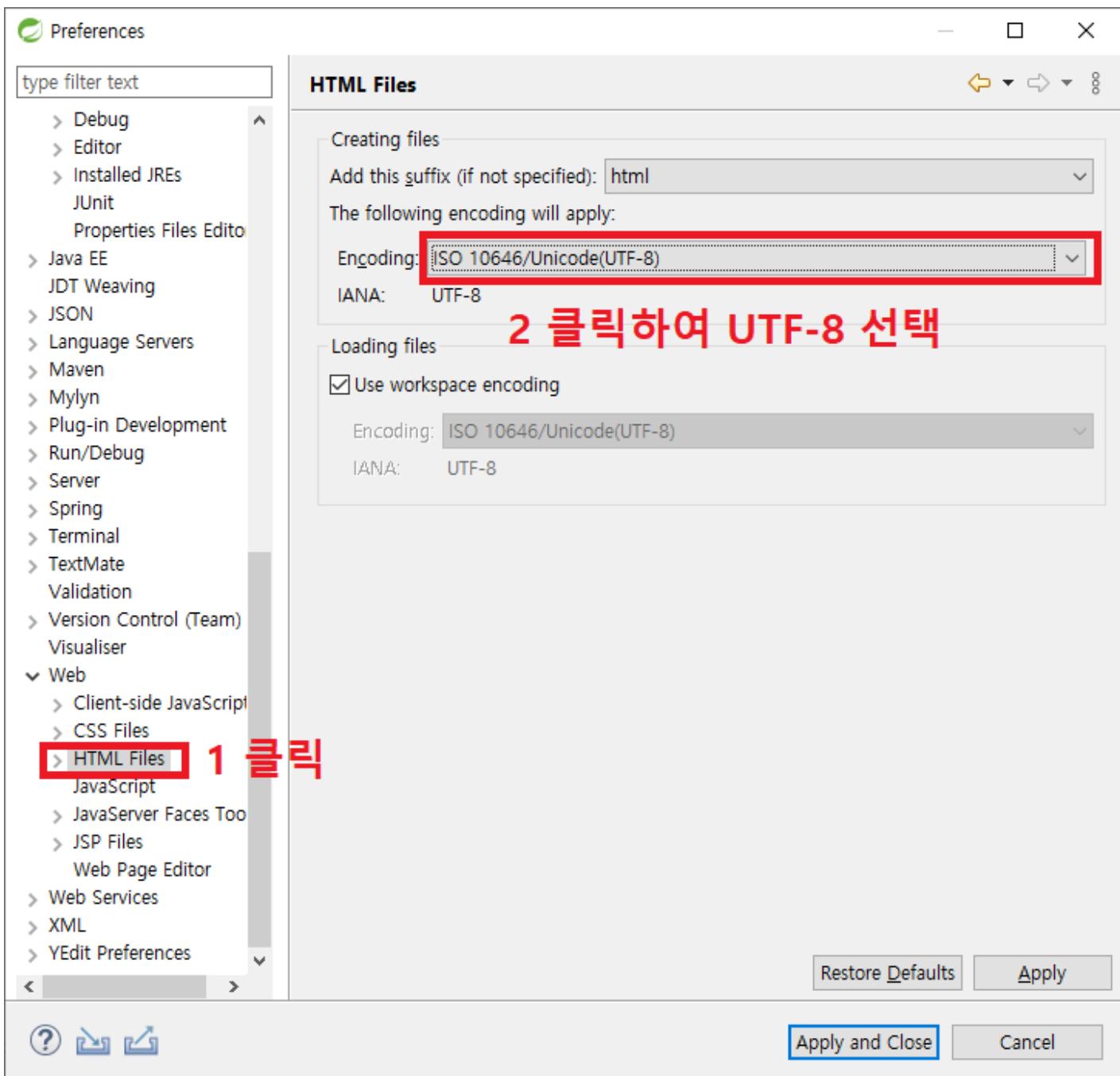
(7) JRE를 설정합니다.



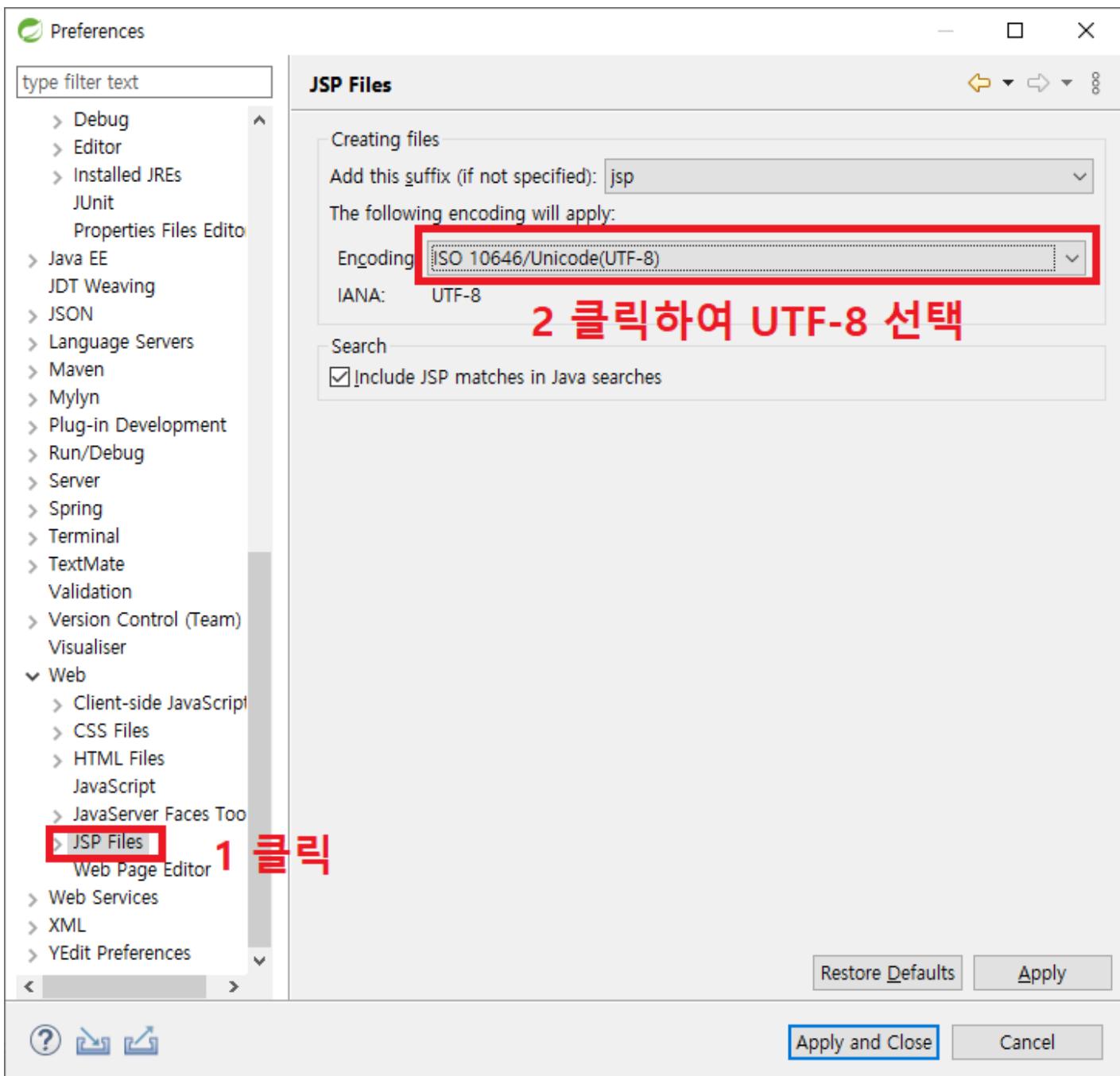
(8) CSS의 인코딩 방식을 설정합니다.



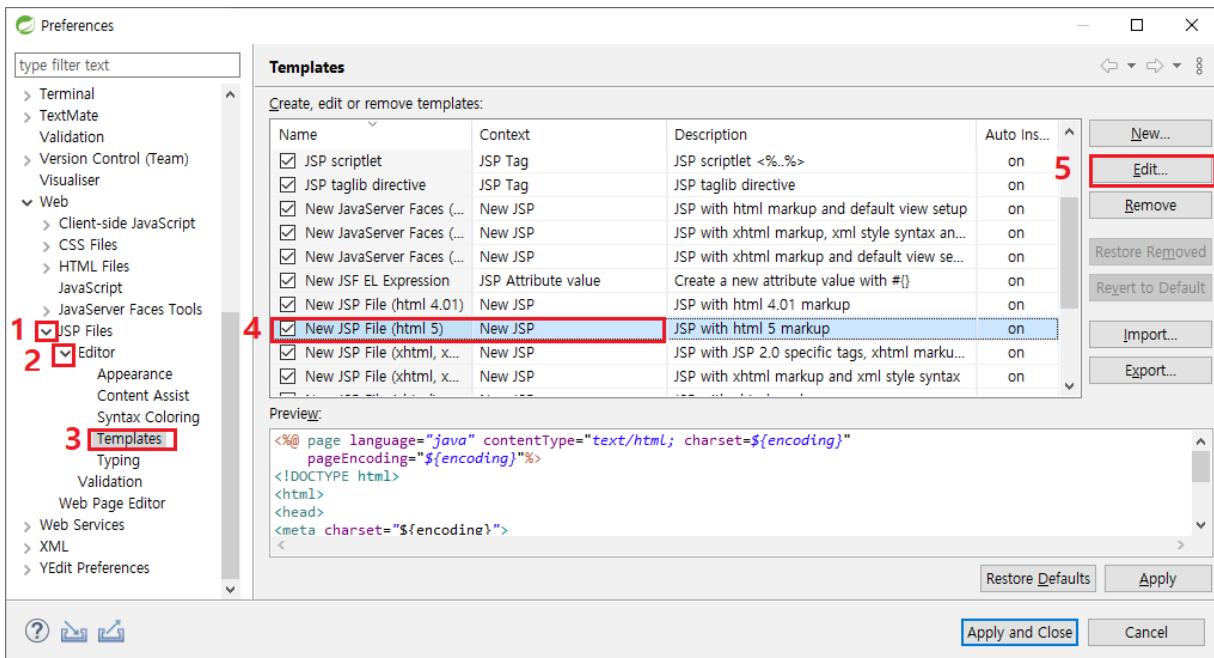
(9) HTML의 인코딩 방식을 설정합니다.



(10) JSP의 인코딩 방식을 설정합니다.

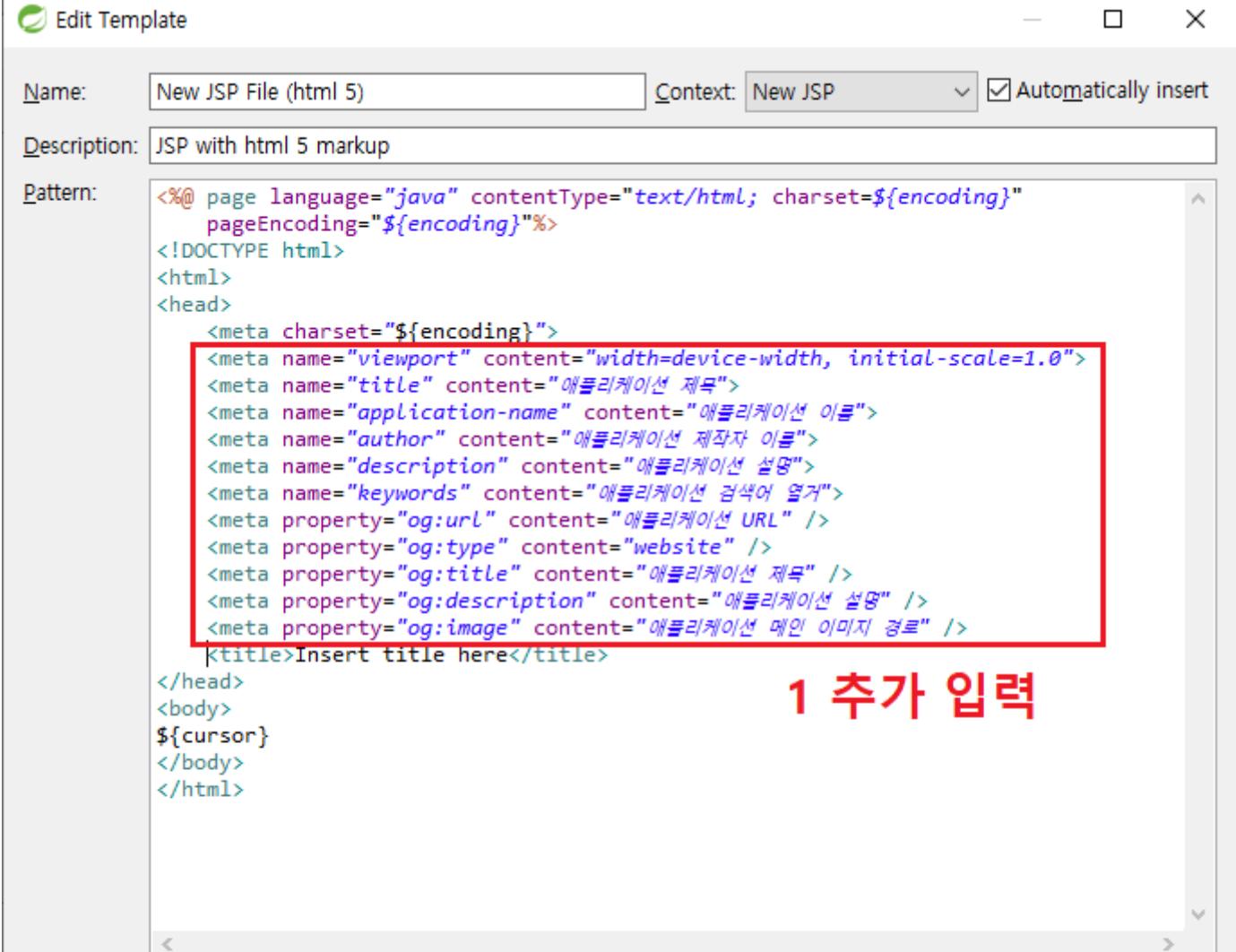


(11) JSP 템플릿을 수정하도록 합니다.



※ JSP 템플릿을 수정하게 되면, 매번 번거롭게 하는 일들을 줄일 수가 있습니다.

## (12) JSP 템플릿의 내용을 수정합니다.

Edit Template

Name: New JSP File (html 5) Context: New JSP  Automatically insert

Description: JSP with html 5 markup

Pattern:

```
<%@ page language="java" contentType="text/html; charset=${encoding}"  
pageEncoding="${encoding}"%>  
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="${encoding}">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <meta name="title" content="애플리케이션 제목">  
    <meta name="application-name" content="애플리케이션 이름">  
    <meta name="author" content="애플리케이션 제작자 이름">  
    <meta name="description" content="애플리케이션 설명">  
    <meta name="keywords" content="애플리케이션 검색어 열거">  
    <meta property="og:url" content="애플리케이션 URL" />  
    <meta property="og:type" content="website" />  
    <meta property="og:title" content="애플리케이션 제목" />  
    <meta property="og:description" content="애플리케이션 설명" />  
    <meta property="og:image" content="애플리케이션 메인 이미지 경로" />  
    <title>Insert title here</title>  
</head>  
<body>  
${cursor}  
</body>  
</html>
```

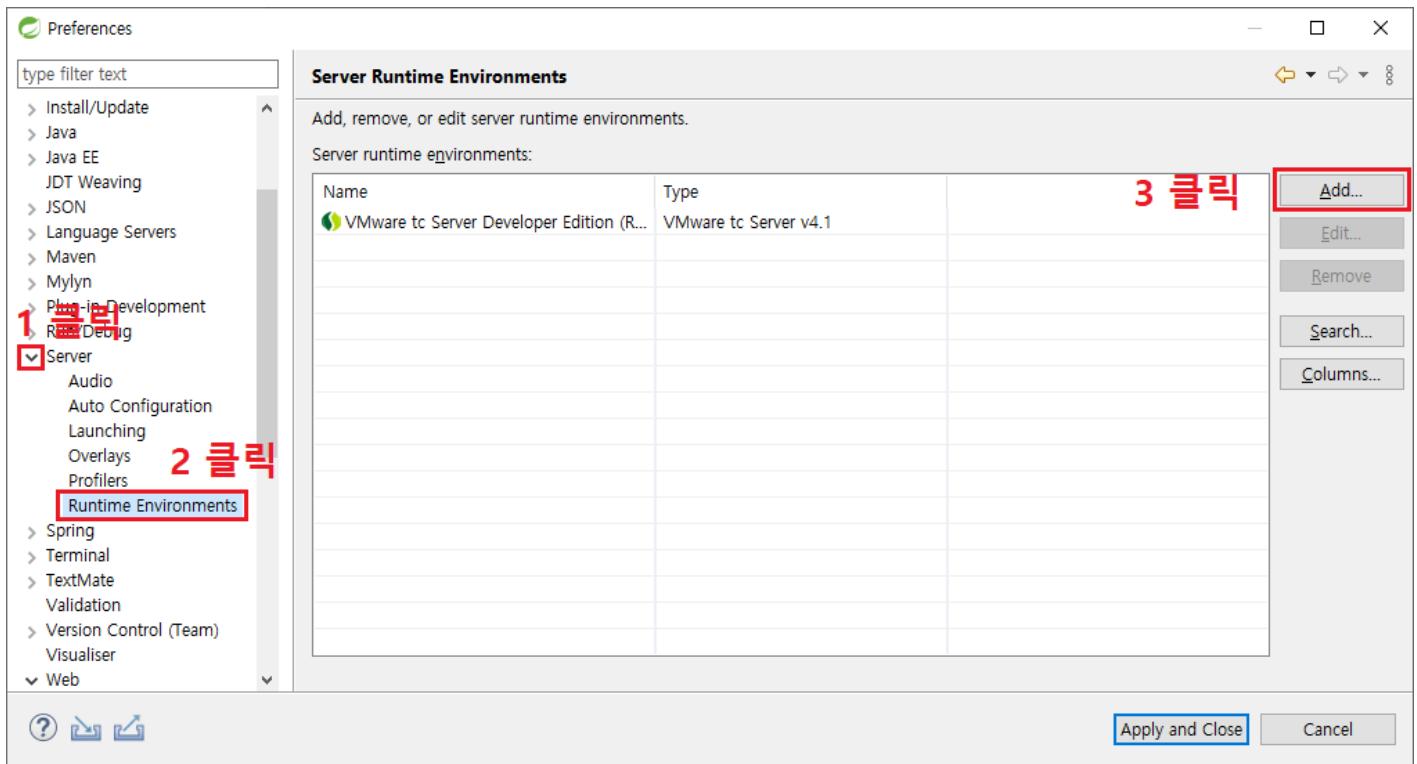
1 추가 입력

2 클릭

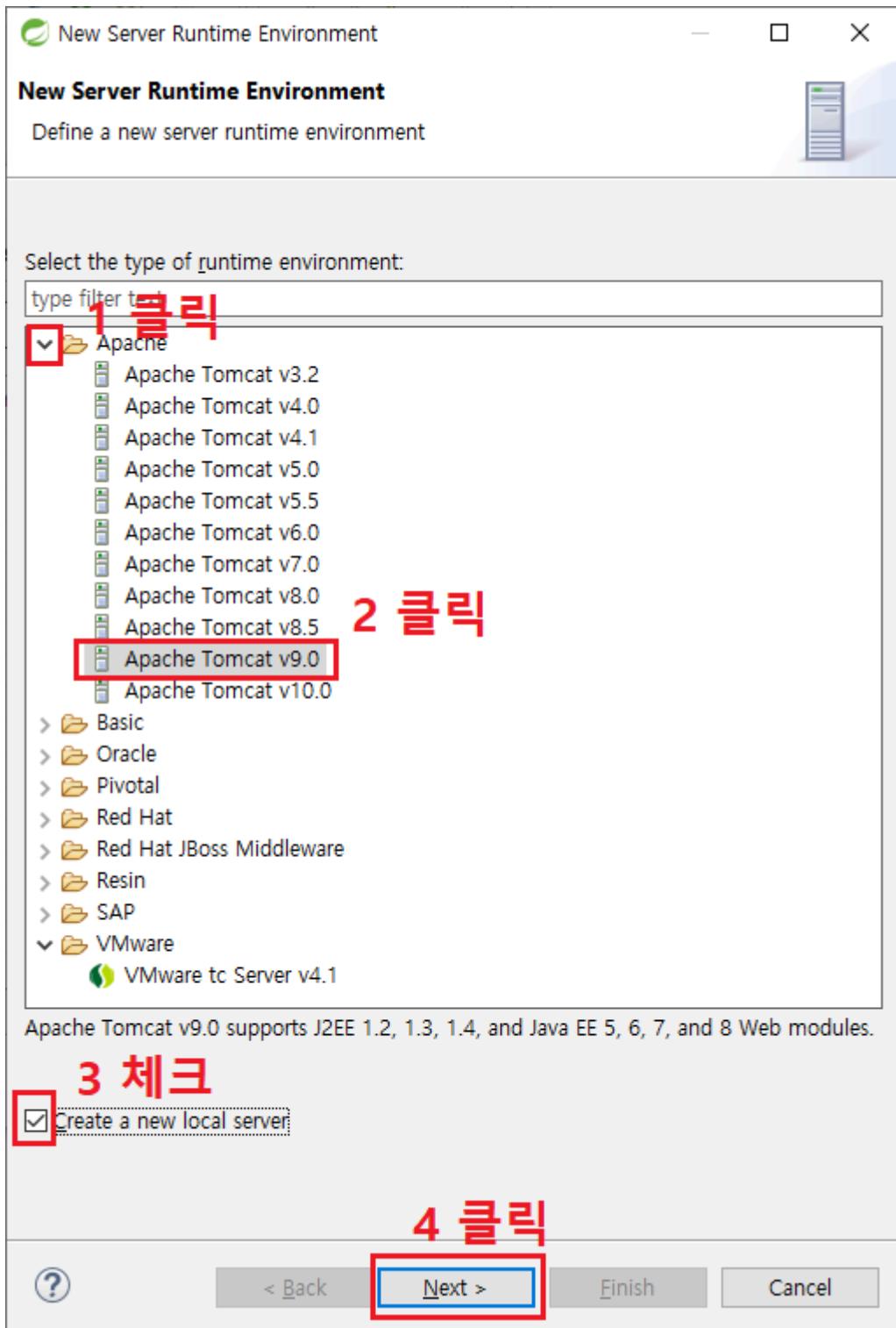
아래는 템플릿에 추가할 내용입니다.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<meta name="title" content="애플리케이션 제목">  
<meta name="application-name" content="애플리케이션 이름">  
<meta name="author" content="애플리케이션 제작자 이름">  
<meta name="description" content="애플리케이션 설명">  
<meta name="keywords" content="애플리케이션 검색어 열거">  
<meta property="og:url" content="애플리케이션 URL" />  
<meta property="og:type" content="website" />  
<meta property="og:title" content="애플리케이션 제목" />  
<meta property="og:description" content="애플리케이션 설명" />  
<meta property="og:image" content="애플리케이션 메인 이미지 경로" />
```

(13) Server 를 설정합니다.

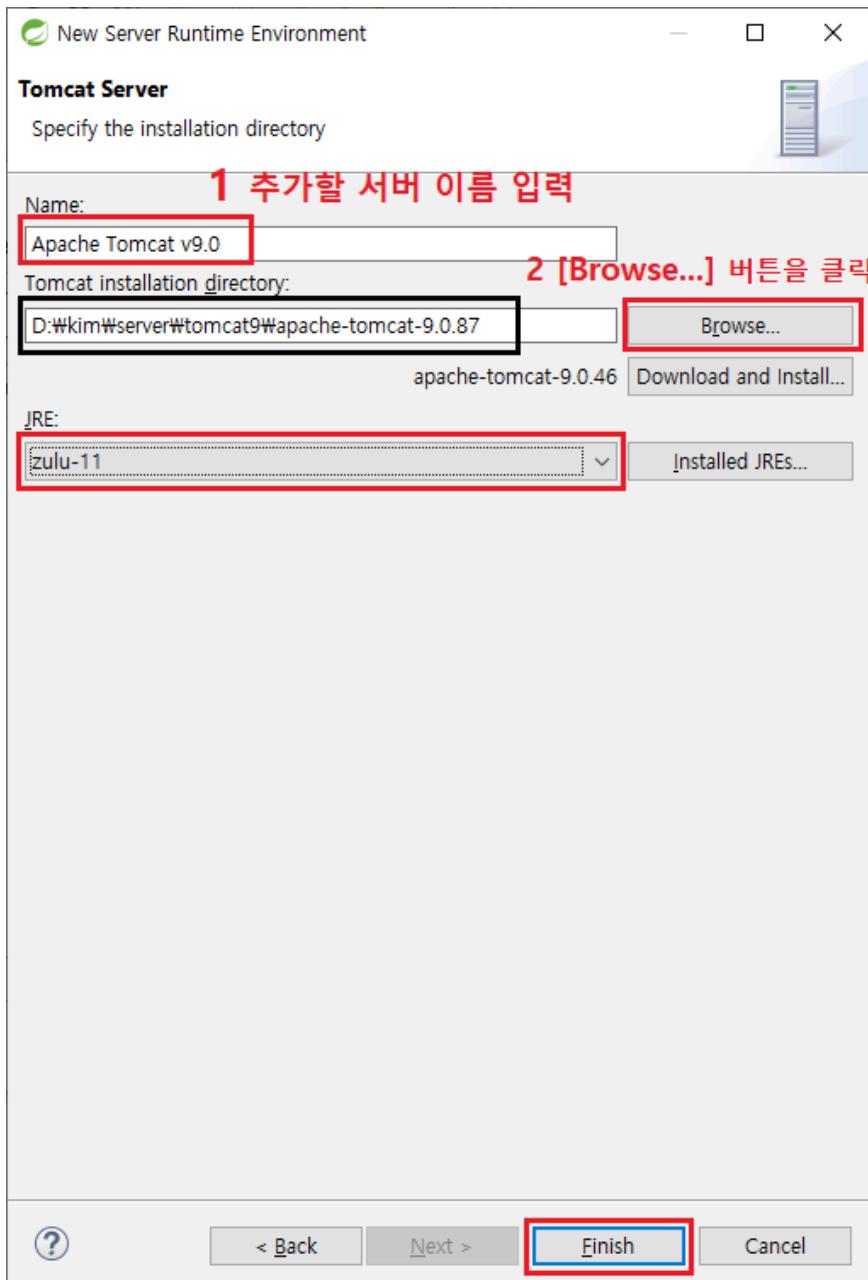


(14) 사용할 Server로 Tomcat Server 를 지정합니다.



※ Create a new local server 항목을 체크하지 않으면, Servers 패널에 나타나지 않으니 꼭 체크해 두시기 바랍니다.

(15) 기존에 설치된 tomcat 9을 Server로 지정합니다.



1 추가할 서버 이름 입력

2 [Browse...] 버튼을 클릭하여 톰캣이 설치된 디렉토리를 지정

### 톰캣 9 다운로드

<https://tomcat.apache.org/download-90.cgi>

해당 페이지에서

64-bit Windows.zip (pgp, sha512) 링크를  
클릭하여 apache-tomcat-9.0.88-windows-  
x64.zip 파일을 다운로드 받아 압축을 해제하  
면 됨

만약, tomcat이 설치되지 않았다면, 아래 작업을 먼저 진행하고, (15)번 작업을 진행하시기 바랍니다.

tomcat.apache.org/download-90.cgi 1. 해당 주소 페이지로 이동



# Apache Tomcat®

Tomcat 9 Software Downloads

Welcome to the Apache Tomcat® 9.x software download page. This page provides download links for obtaining the latest version of Tomcat 9.0.x software, as well as links to the archives of older releases.

Unsure which version you need? Specification versions implemented, minimum Java version required and lots more useful information may be found on the [which version?](#) page.

**Quick Navigation**

[KEYS](#) | [9.0.88](#) | [Browse](#) | [Archives](#)

**Release Integrity**

You must [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file which contains the OpenPGP keys of Tomcat's Release Managers. We also provide SHA-512 checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

**Mirrors**

You are currently using <https://dlcdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are backup mirrors (at the end of the mirrors list) that should be available.

Other mirrors: <https://dlcdn.apache.org/> ▾ Change

**9.0.88**

Please see the [README](#) file for packaging information. It explains what every distribution contains.

**Binary Distributions**

- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows zip \(pgp, sha512\)](#)
  - [64-bit Windows zip \(pgp, sha512\)](#) 2. 해당 링크를 클릭하여 zip 파일 다운로드
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
  - [tar.gz \(pgp, sha512\)](#)
- Deployer:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
- Embedded:
  - [tar.gz \(pgp, sha512\)](#)
  - [zip \(pgp, sha512\)](#)

**Source Code Distributions**

## 아파치톰캣 다운로드 페이지 TOMCAT9

다운로드 받은 apache-tomcat-9.0.88-windows-x64.zip 파일을 원하는 곳으로 이동 후 압축을 해제합니다.

(16) 불필요한 VMWare Server를 제거하고, sts 설정을 종료합니다.

Preferences

type filter text

**Server Runtime Environments**

Add, remove, or edit server runtime environments.

Server runtime environments:

Name	Type
Apache Tomcat v9.0	Apache Tomcat v9.0 1 클릭
VMware tc Server Developer Edition (R...)	VMware tc Server v4.1

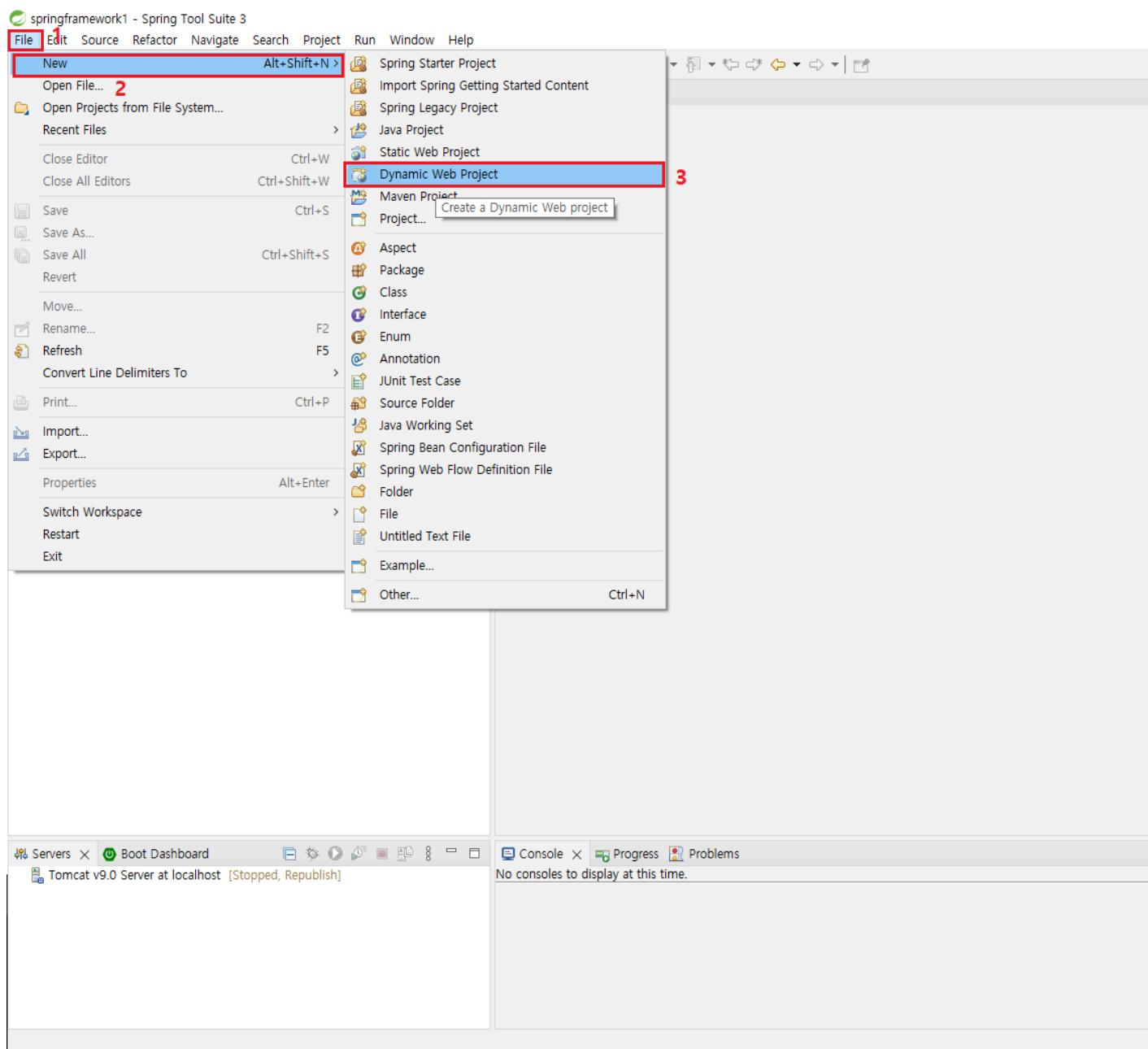
1 클릭  
2 클릭하여  
불필요한 서버 제거  
3 클릭하여 변경 내용 모두 적용하고 닫기

D:/kim/sts/sts/sts-bundle/pivotal-tc-server

Apply and Close Cancel

## 1-4. Dynamic Web Project 를 Maven Project로 변환하기

### (1) Dynamic Web Project 를 새롭게 추가합니다.



### (2) Project 의 기본 정보를 입력하고 설정합니다.

New Dynamic Web Project

**Dynamic Web Project**

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: **webPro1** 1 프로젝트 이름 입력

Project location  
 Use default location  
Location: D:\kim\springframework1\webPro1 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경 Browse...

Target runtime  
Apache Tomcat v9.0 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경 New Runtime...

Dynamic web module version  
4.0 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경

Configuration  
Default Configuration for Apache Tomcat v9.0 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경 Modify...

A good starting point for working with Apache Tomcat v9.0 runtime. Additional facets can later be installed to add new functionality to the project.

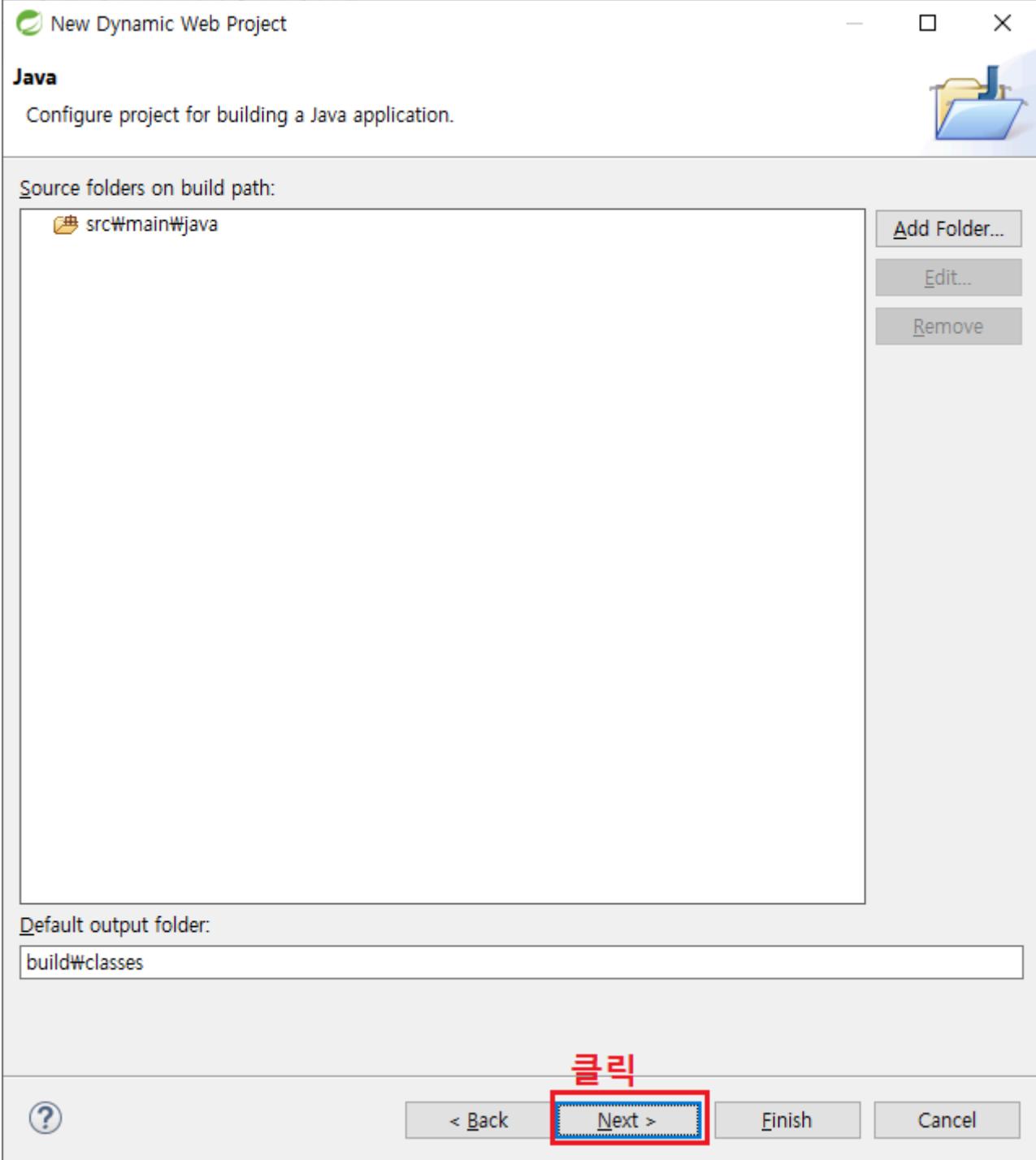
EAR membership  
 Add project to an EAR  
EAR project name: EAR 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경 New Project...

Working sets  
 Add project to working sets 밑줄 친 부분은 확인하고, 맞지 않는 경우 변경 New... Select...

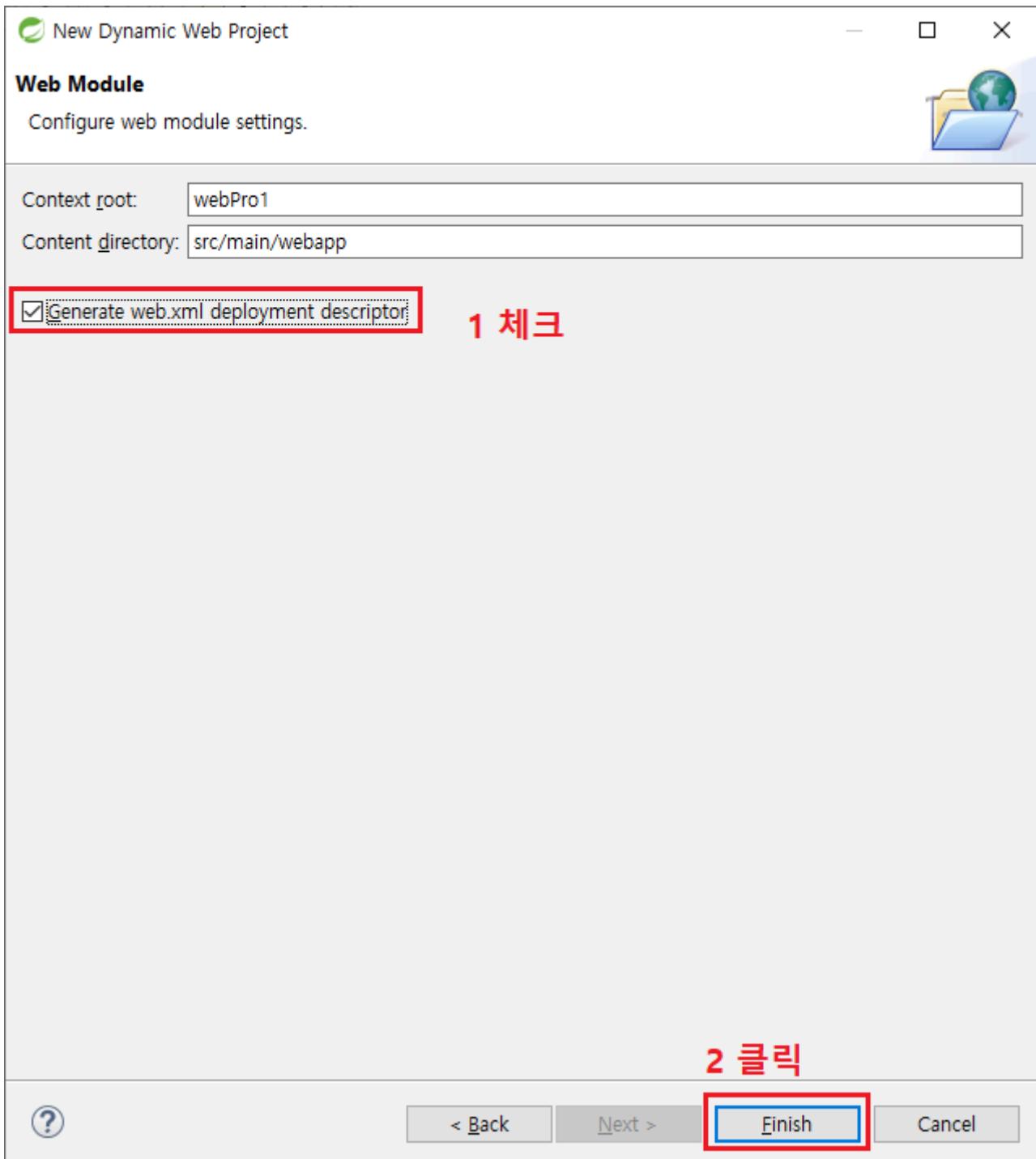
Working sets: 2 버튼 클릭

Next > Finish Cancel

(3) Project에서 활용될 디렉토리 설정과 배포할 파일이 존재할 출력 디렉토리를 지정합니다.

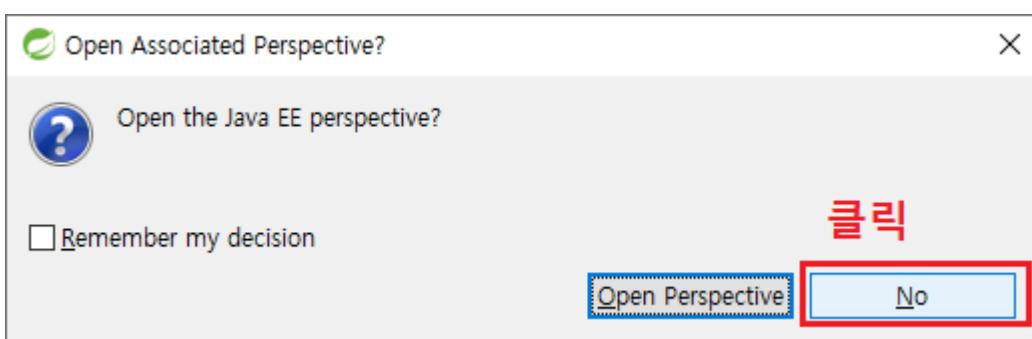


(4) Project에서 사용할 웹 모듈과 설정 방식을 지정합니다.

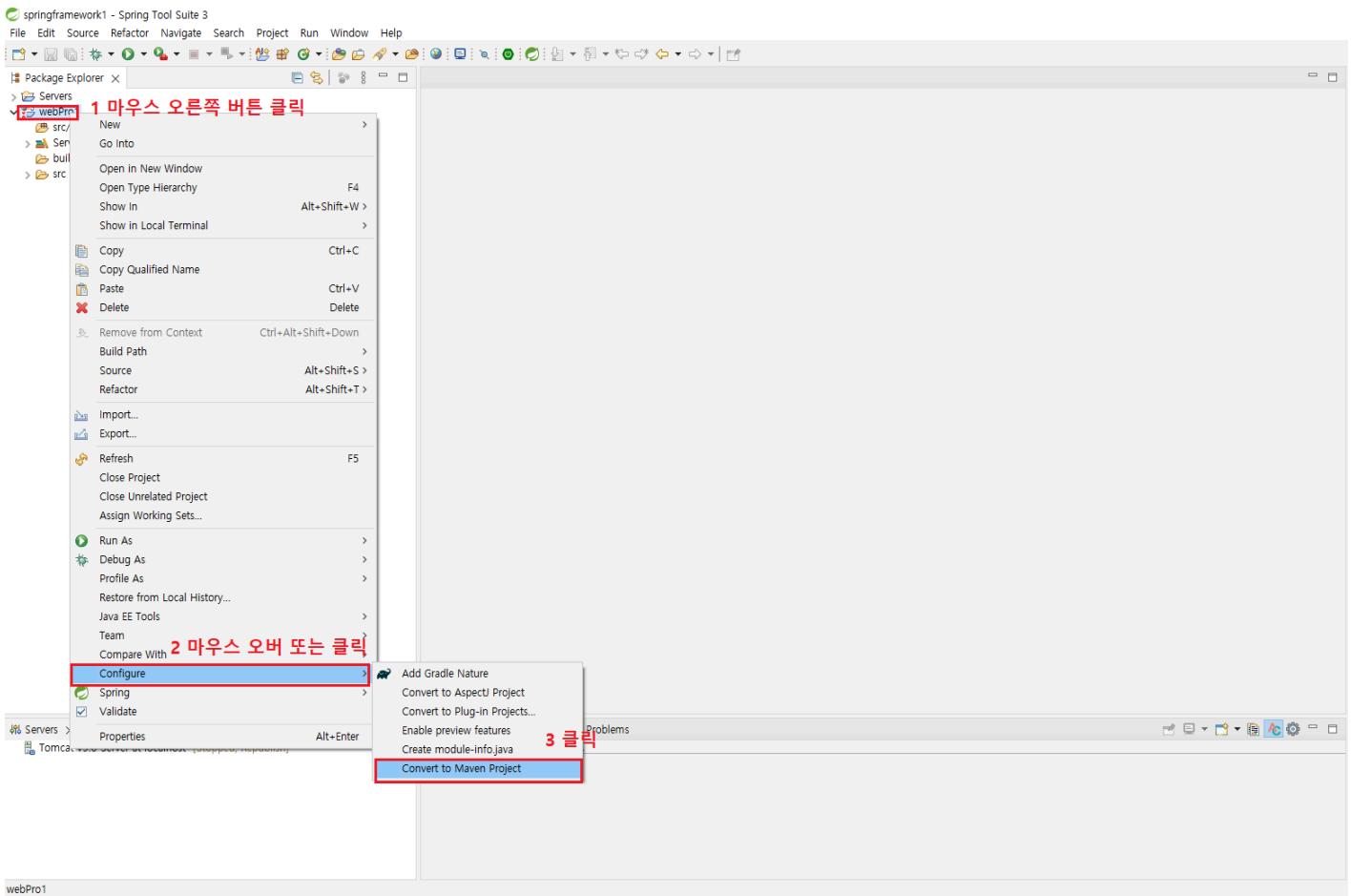


※ Create web.xml deployment descriptor 항목을 체크하지 않으면, 수동으로 만드는 수고를 해야 하니 꼭 체크하시기 바랍니다.

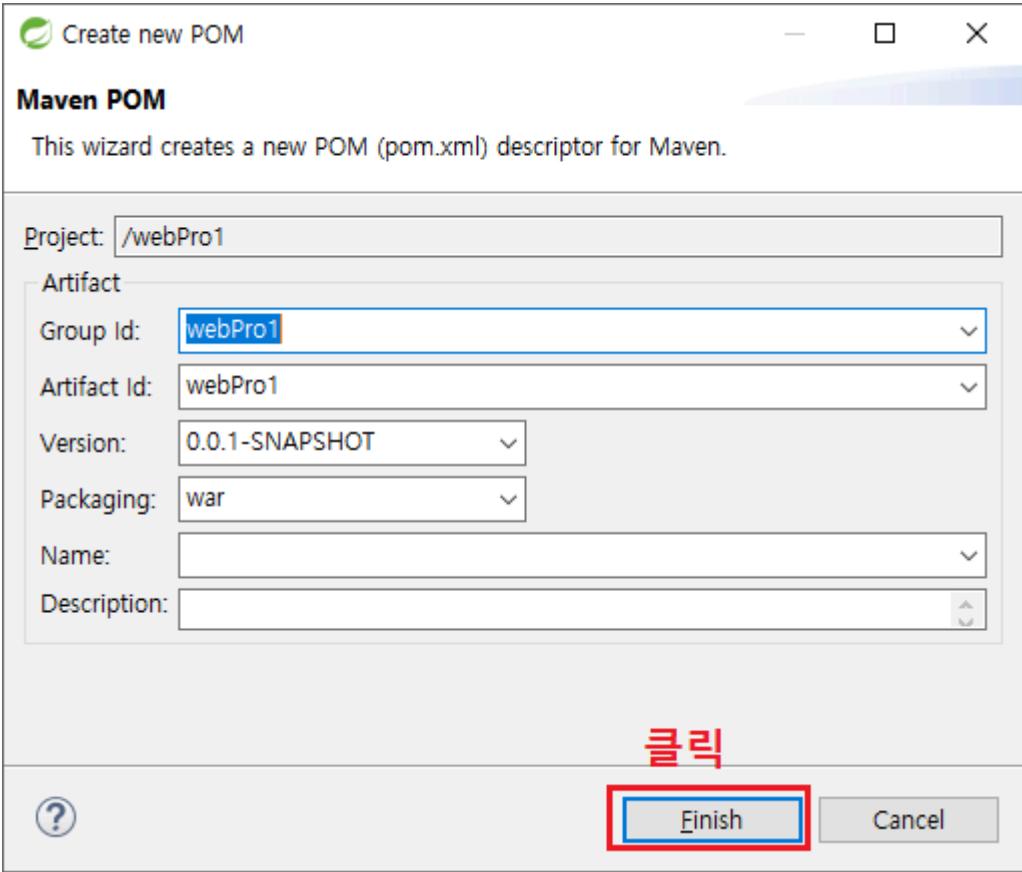
(5) sts에서 Dynamic Web Project를 추가하게 되면, JSP/Servlet 개발 환경에 적합한 화면으로 변경하겠다는 메시지가 출력됩니다.



## (6) Dynamic Web Project를 Maven Project로 변환합니다.

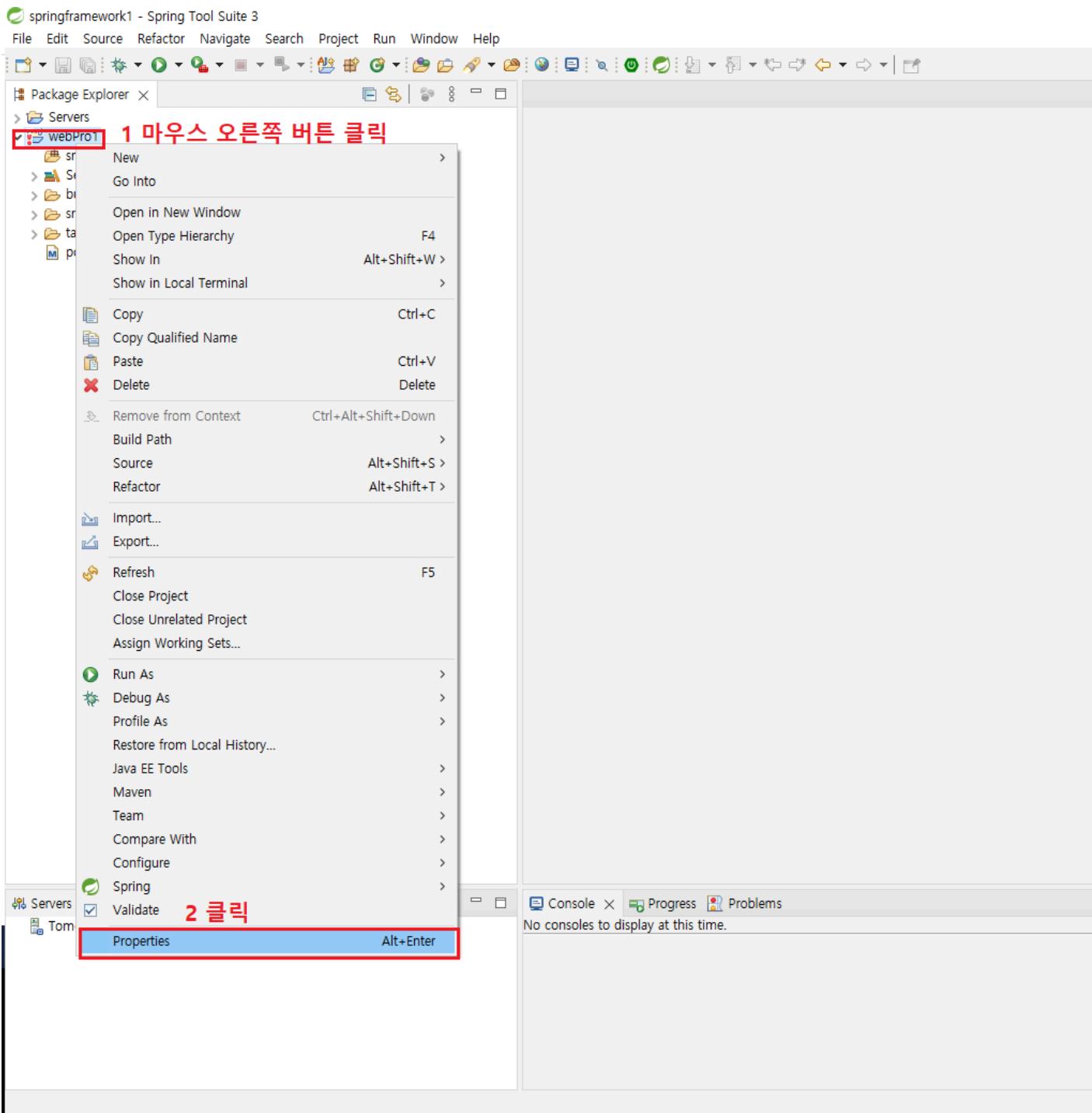


## (7) Maven Project로 변환시 프로젝트의 그룹아이디와 저작아이디 그리고, 패키징 방식 등을 지정합니다.

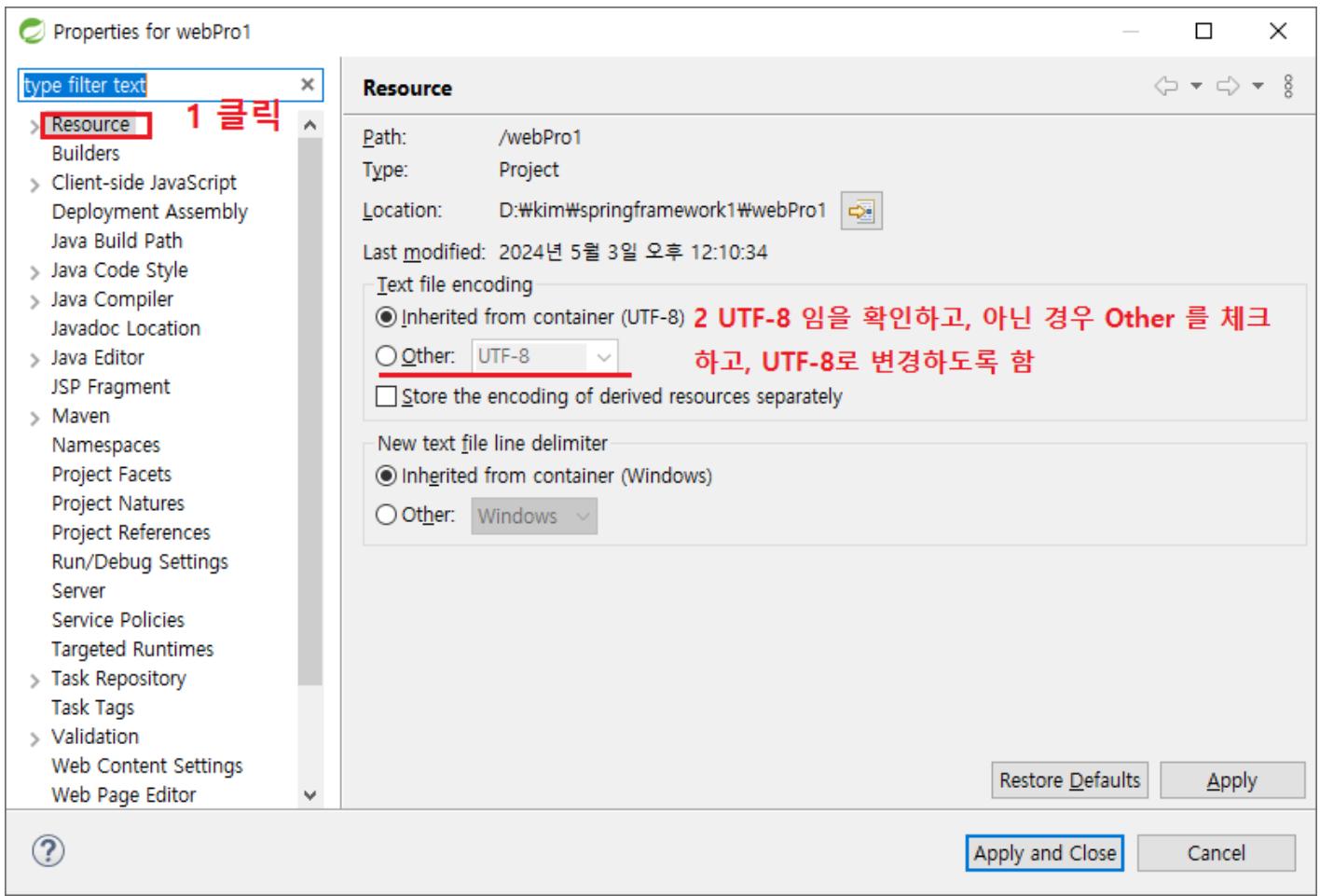


## 1-5. 프로젝트 환경 설정하기

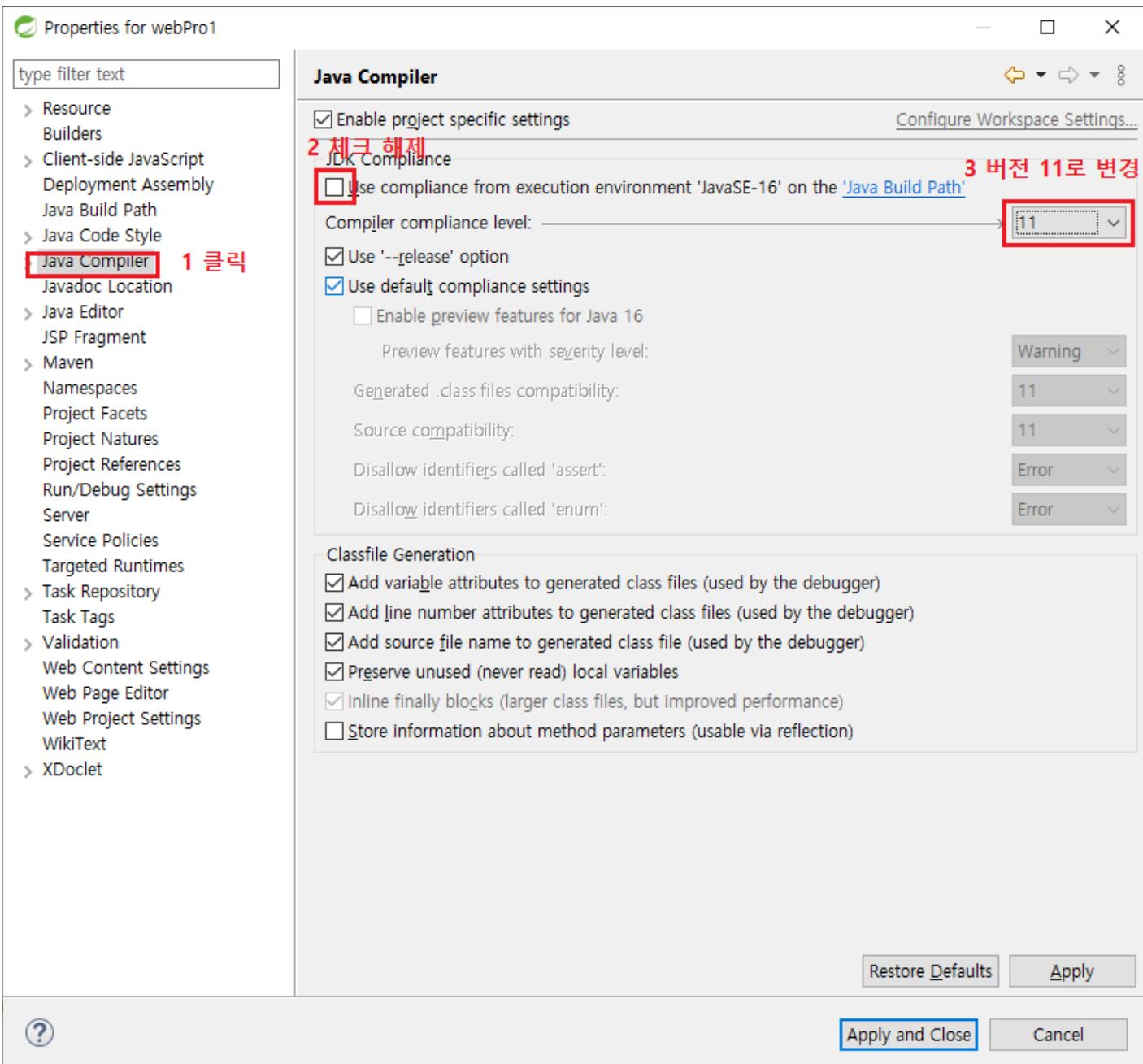
(1) 프로젝트 환경을 설정창을 불러옵니다.



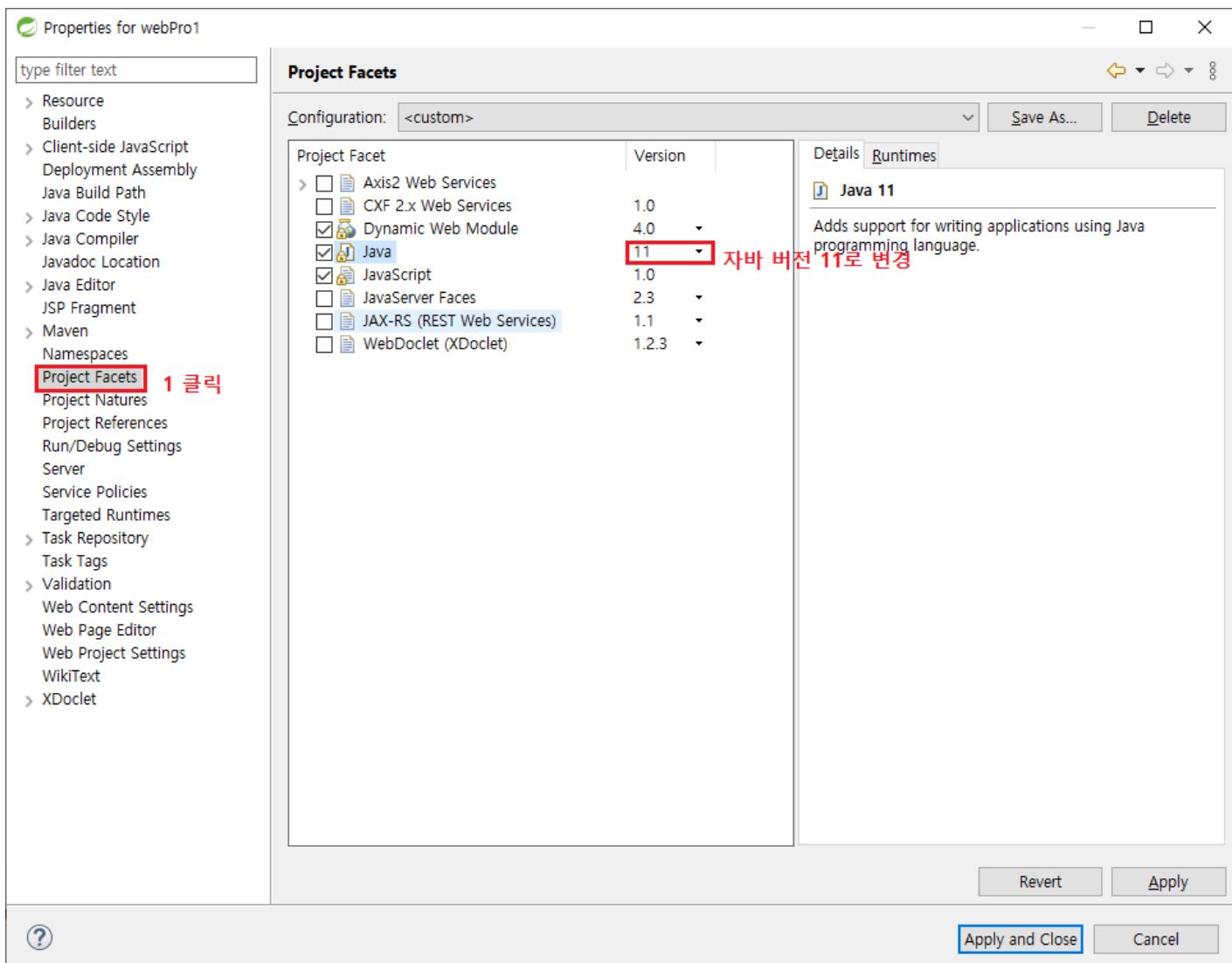
(2) 프로젝트의 인코딩 방식을 지정합니다.



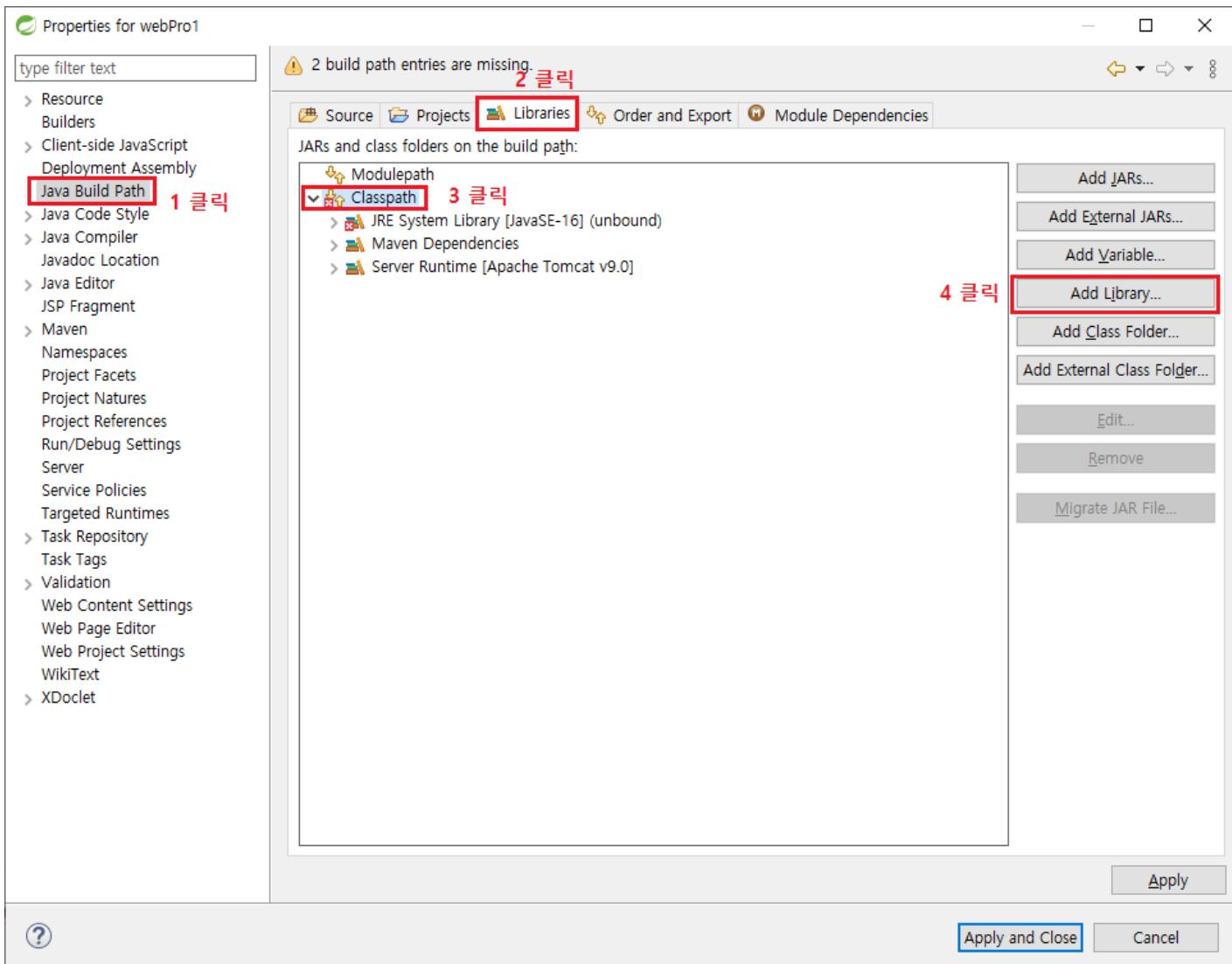
(3) 프로젝트에서 사용할 자바 컴파일러를 설정합니다.



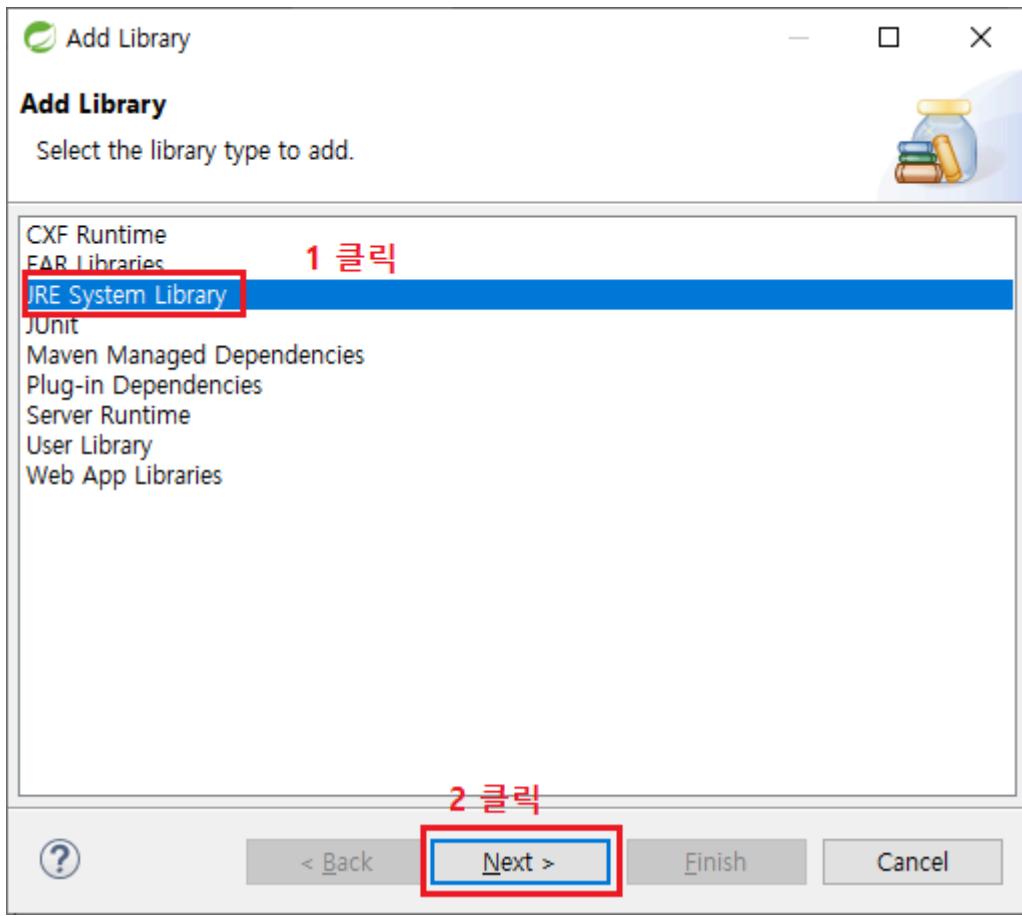
(4) 프로젝트의 프론트엔드/백엔드 웹 환경을 설정합니다.



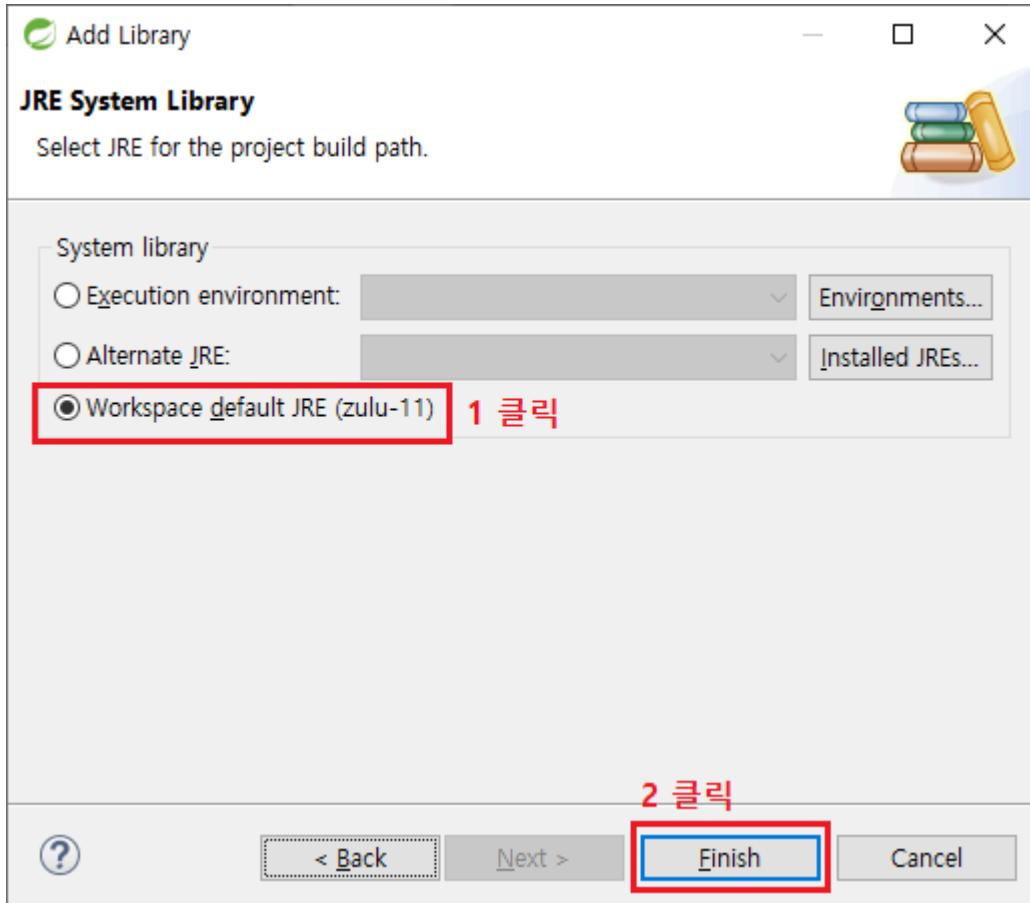
(5) 프로젝트에서 사용할 JRE 환경을 지정합니다.



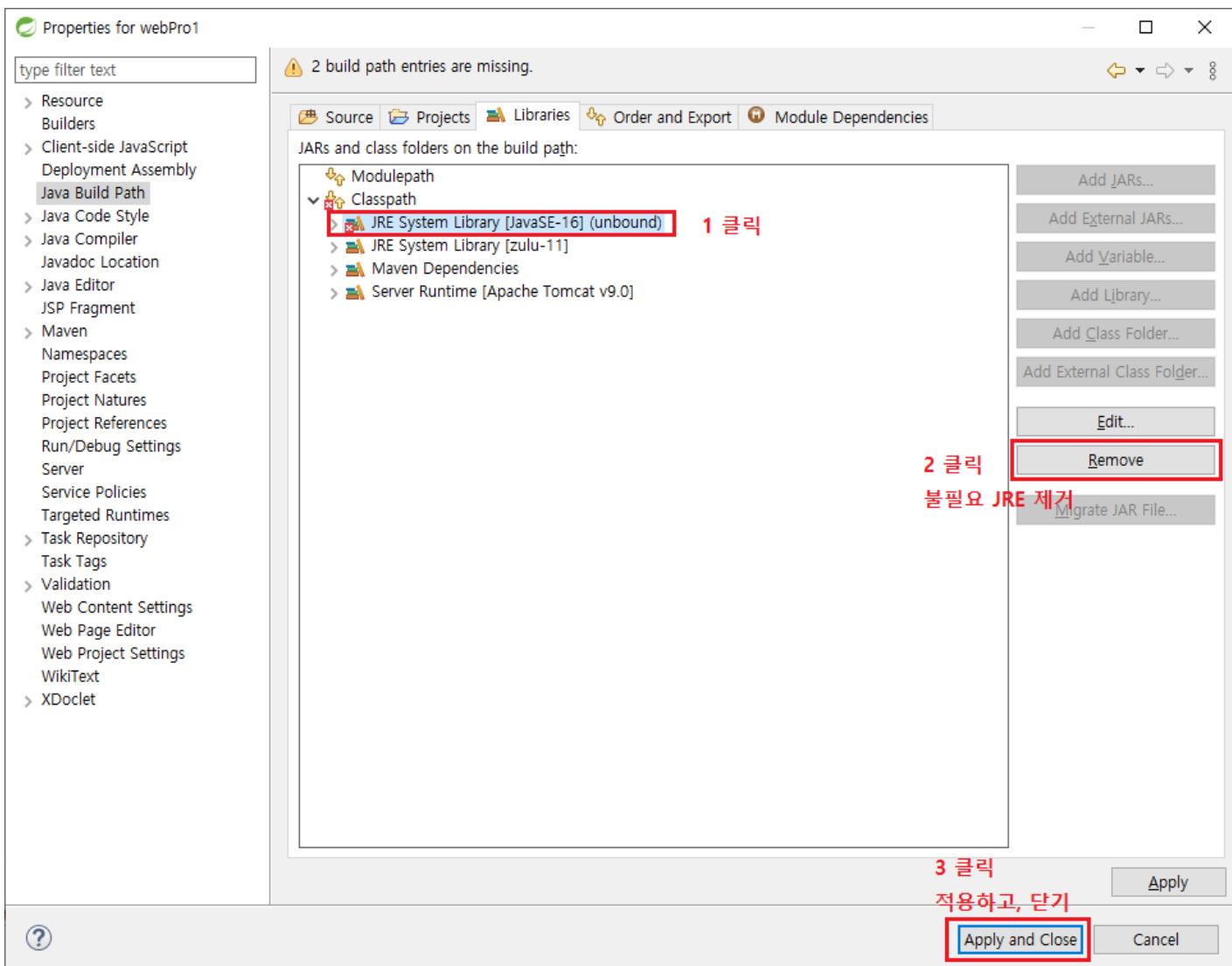
(6) 프로젝트에서 사용할 JRE 환경을 불러옵니다.



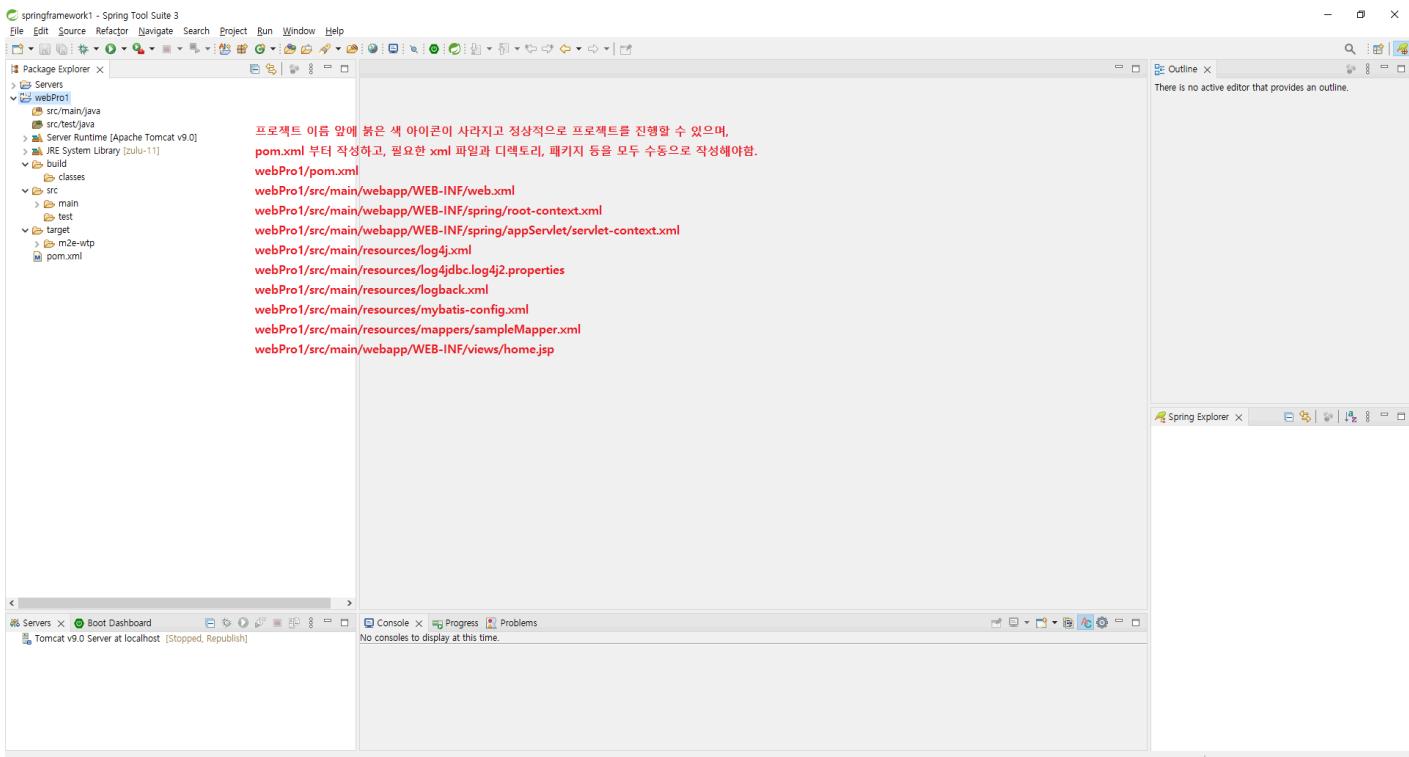
(7) 프로젝트에서 사용할 JRE 환경을 선택합니다.



(8) 프로젝트에서 사용하지 않는 JRE 환경을 제거하고, 환경 설정을 종료합니다.



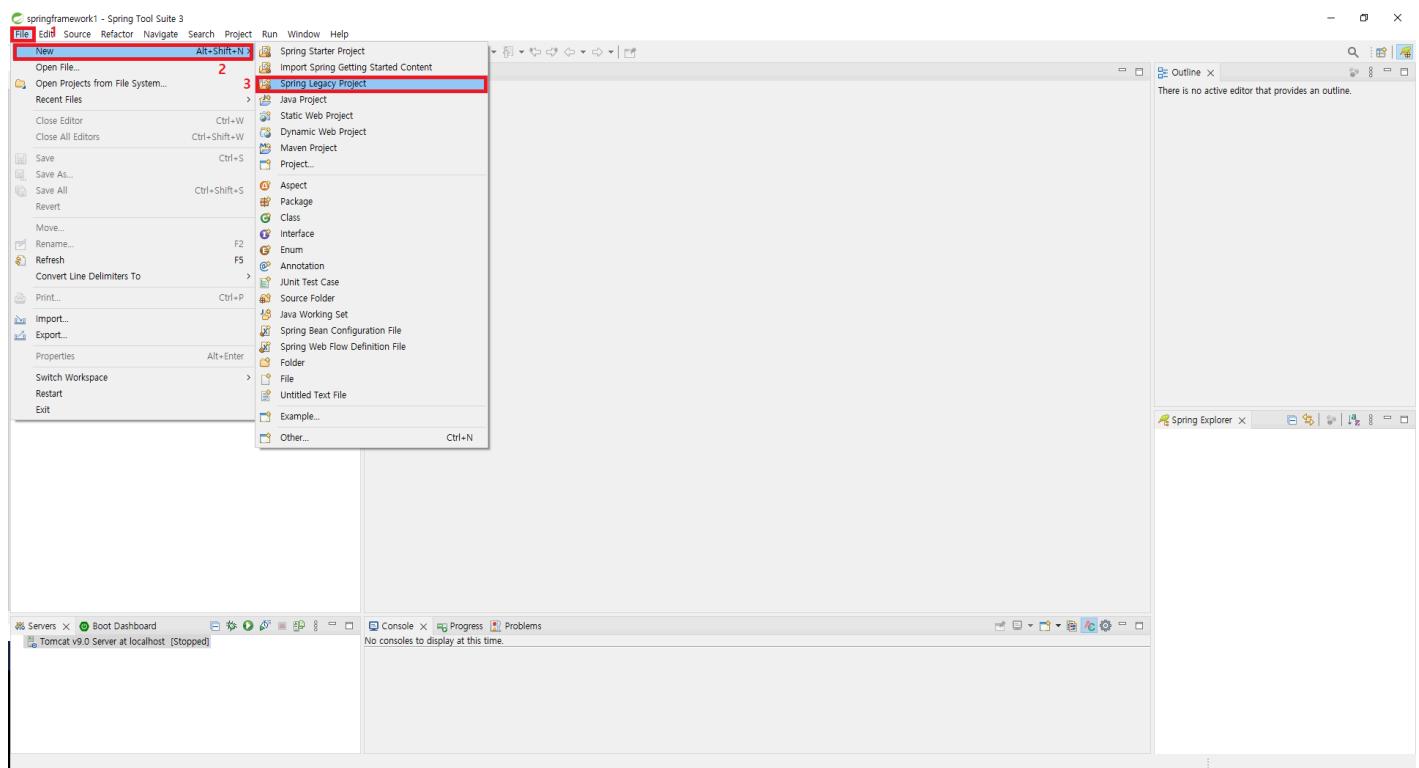
## (9) 프로젝트 개발작업을 시작합니다.



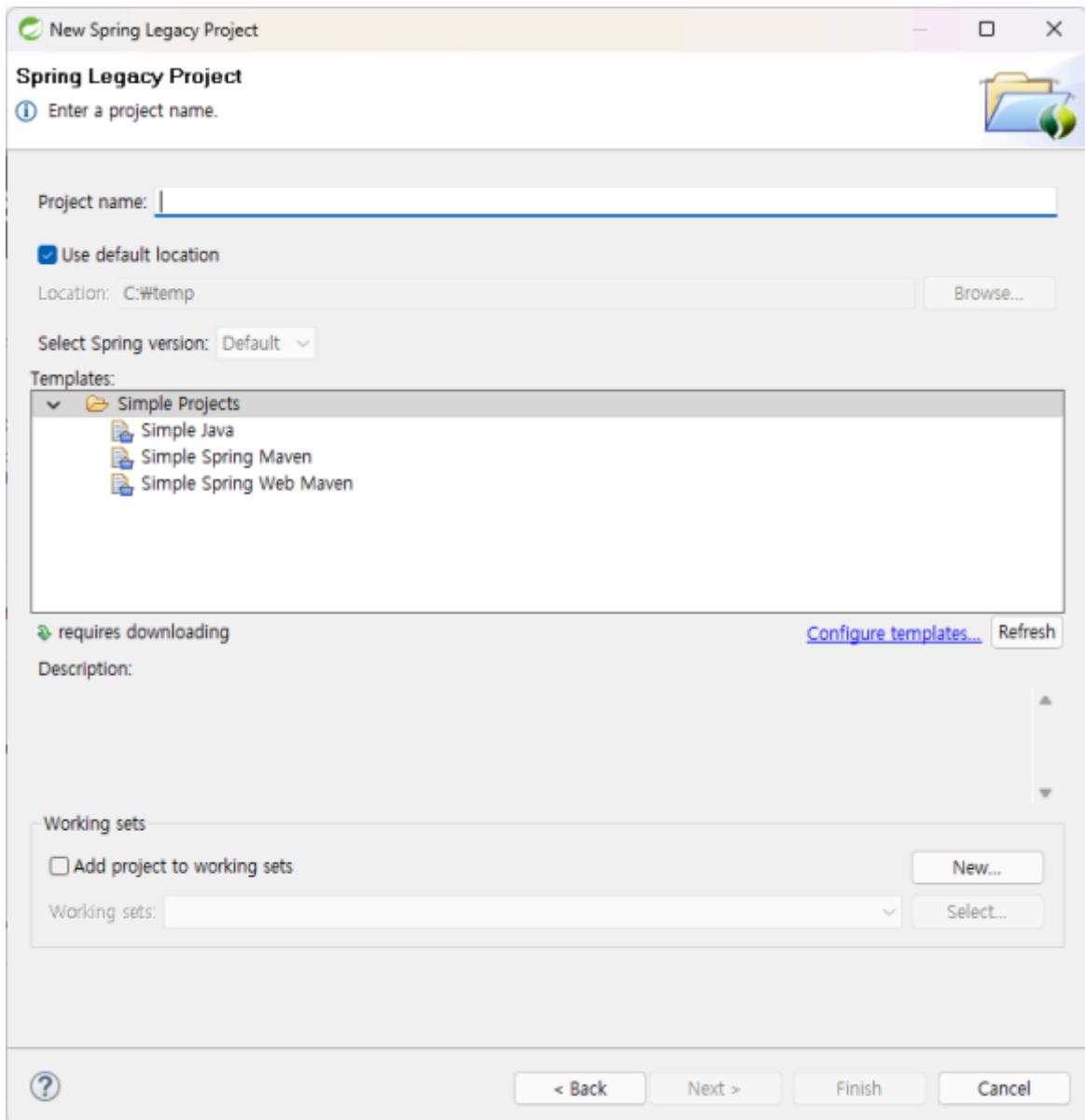
※ 위와 같이 Dynamic Web Project에서 Spring Legacy Project 변환하여 작성하게 되면, 모두 수동 설정 작업을 해야 하므로, 상당히 설정에 너무 시간이 소요된다. 그래서 이번에는 아래 화면들은 Spring Legacy Project 직접 만드는 것이 편합니다.

## 1-6. sts에 Spring Legacy Project 템플릿 추가하기

(1) Spring Legacy Project 를 새롭게 추가합니다.



(2) 현재는 Spring Legacy Project 템플릿이 없는 상황입니다.

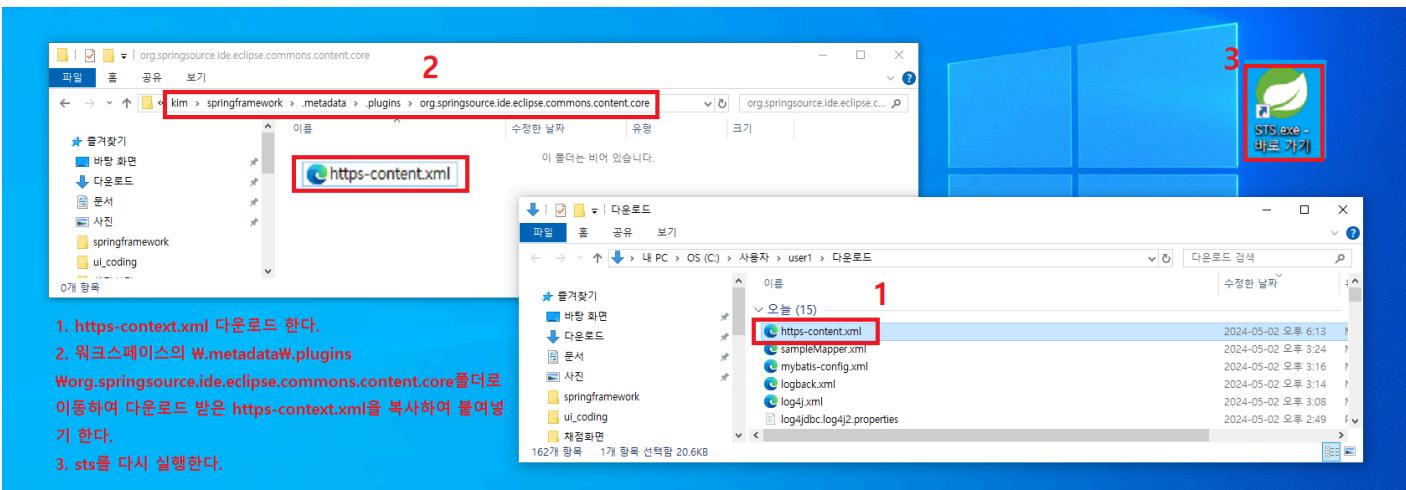


(3) 새로운 템플릿 추가작업을 위해 sts를 종료합니다.

그런데 위에서 본바와 같이 2024년 되자 sts에서 Spring Legacy Project 를 만들 수 있는 템플릿이 사라졌다. 그래서 별도의 템플릿 추가 작업을 진행하여 프로젝트를 생성해야 합니다.

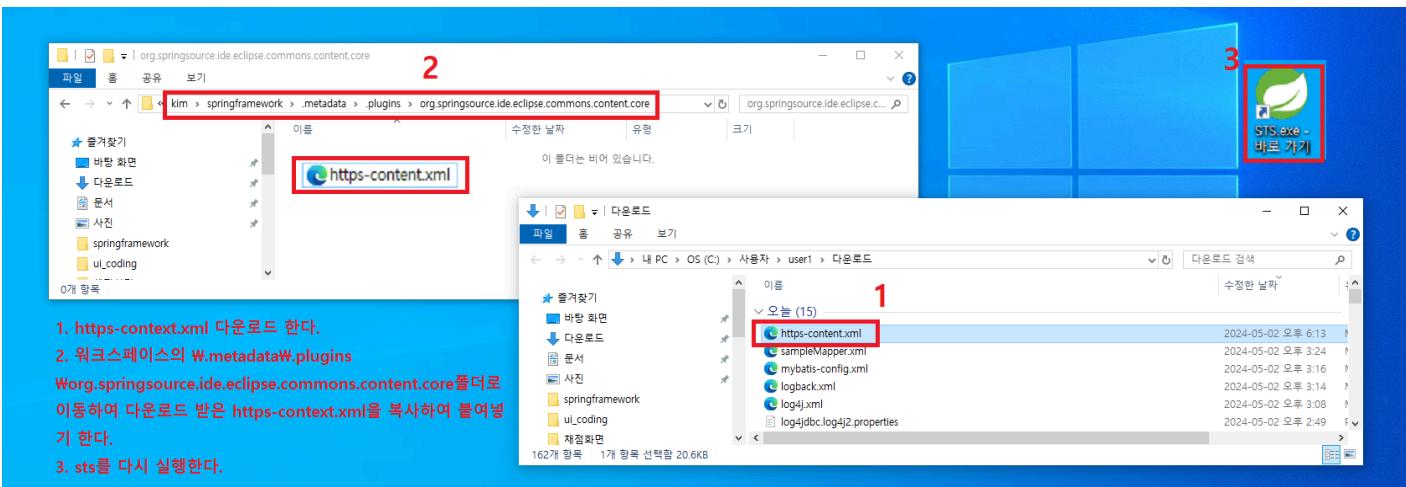
<https://content.xml> 다운로드 링크 ↪ <https://content.xml> 다운로드 받기

(4) Spring Legacy Project 템플릿 추가 작업을 진행합니다.

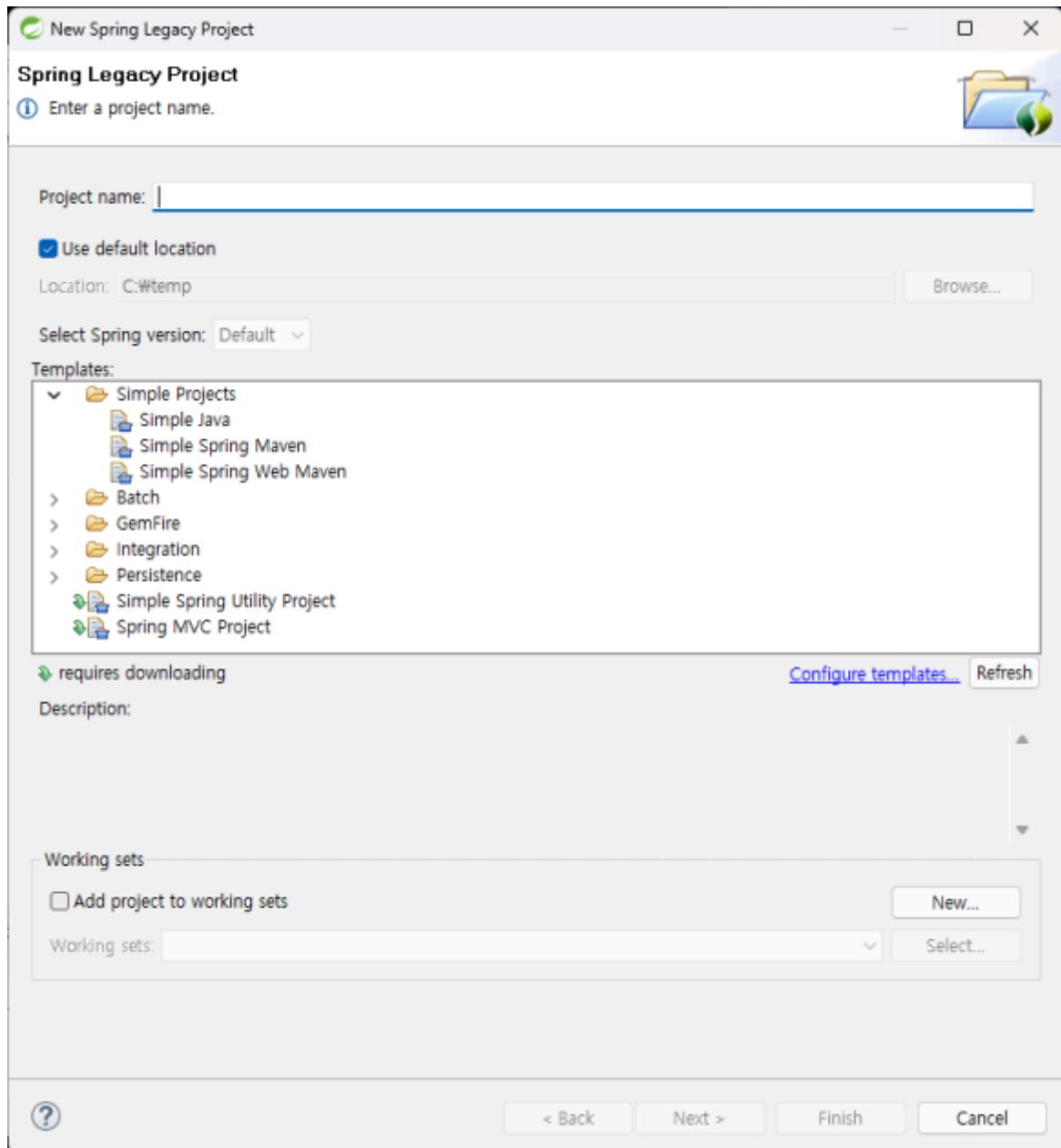


## (5) sts를 다시 재기동합니다.

## (6) Spring Legacy Project 를 새롭게 추가합니다.

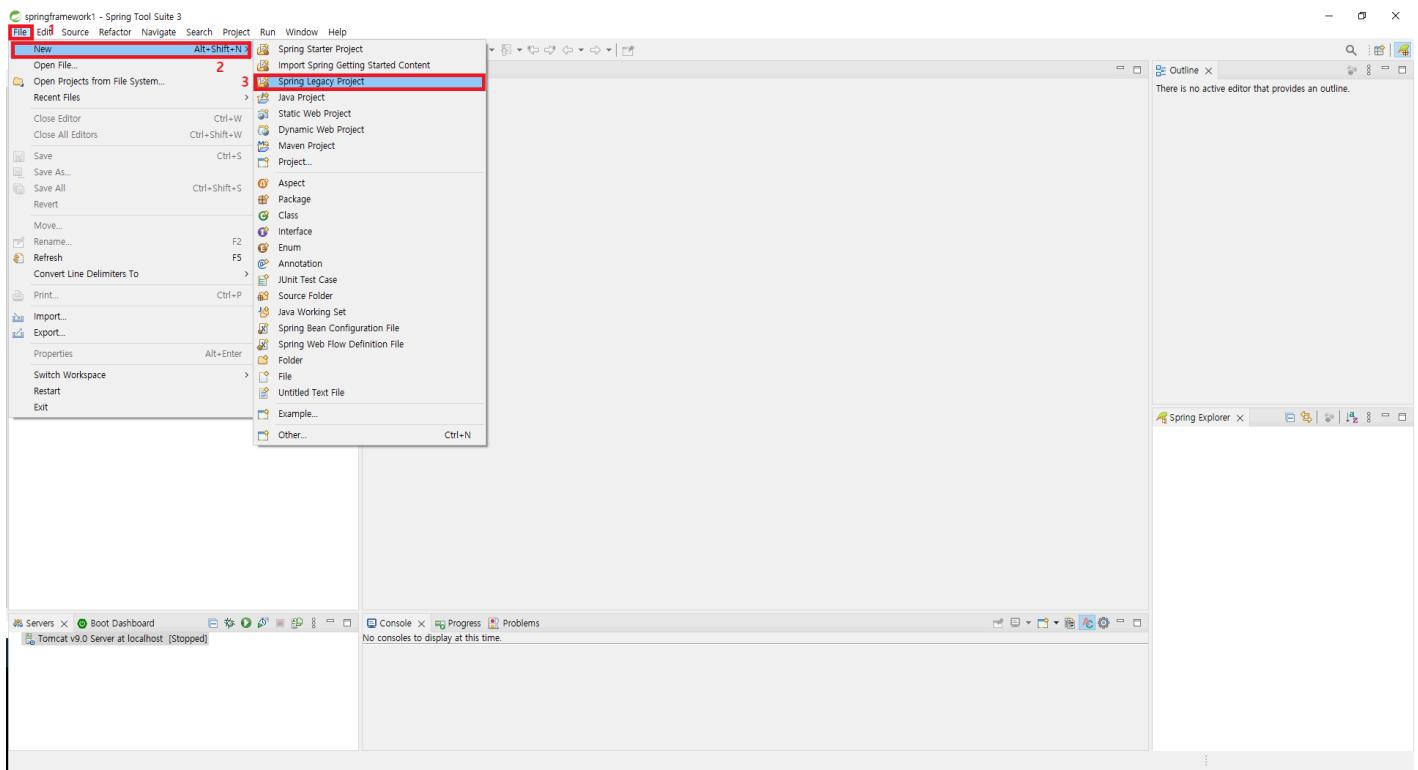


## (7) 화면 처럼 없던 템플릿이 추가된 것을 확인합니다.

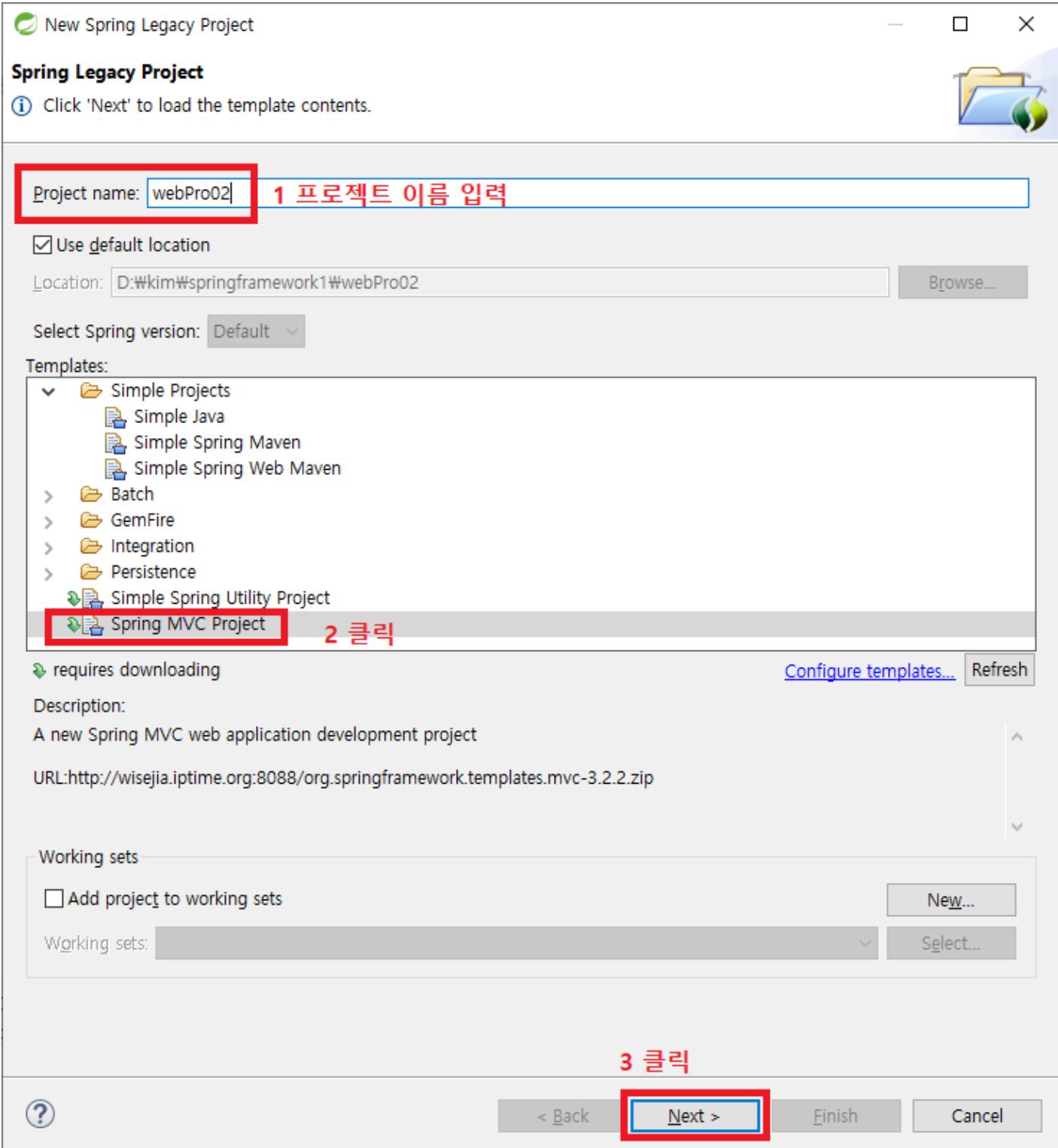


## 1-7. sts에서 Spring Legacy Project 바로 추가하기

(1) 메뉴에서 Spring Legacy Project를 새롭게 추가합니다.



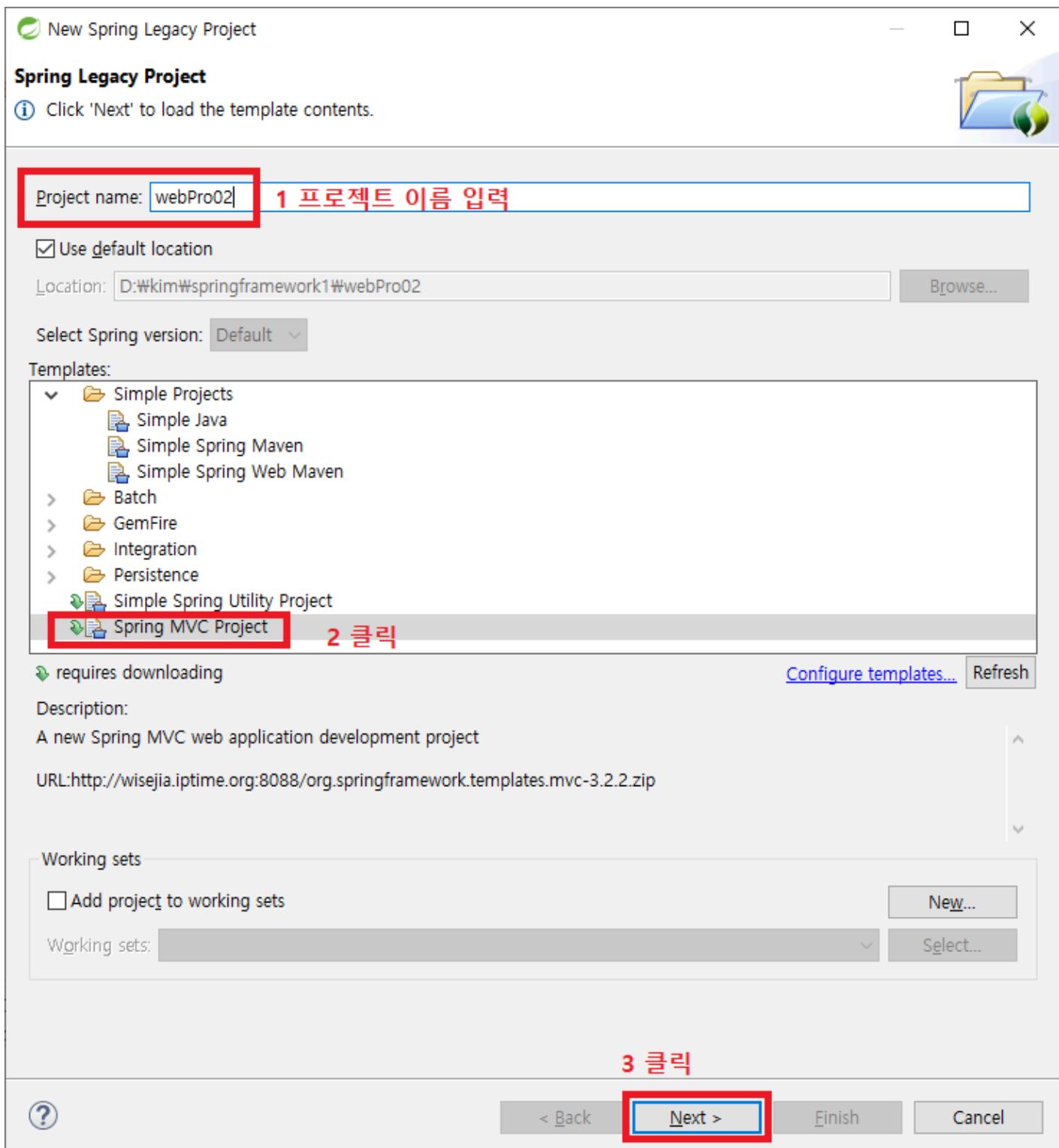
(2) 프로젝트의 이름을 입력하고, Spring MVC Project를 선택합니다.



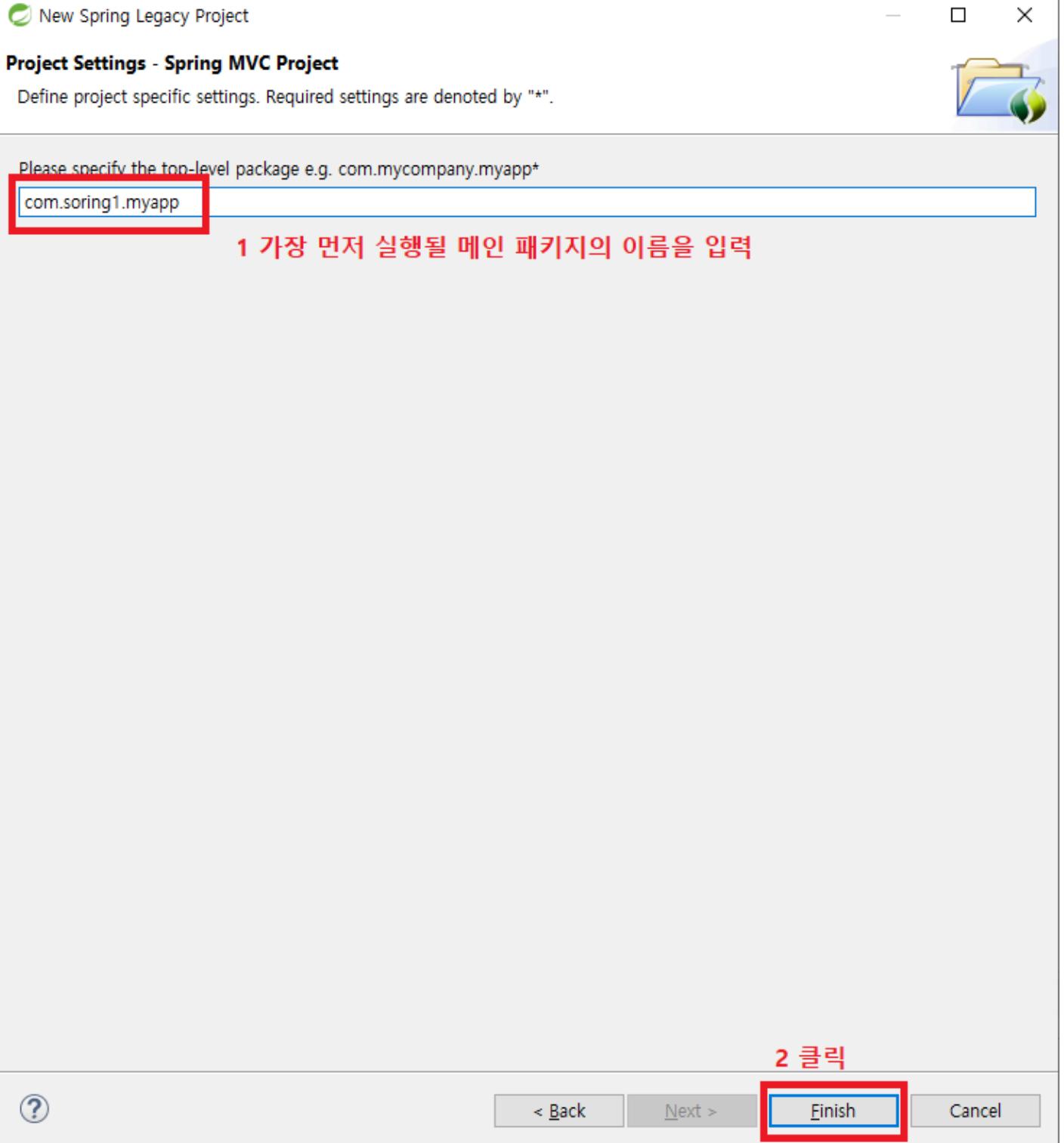
(3) 메시지 창에서 [Yes] 버튼을 누릅니다.



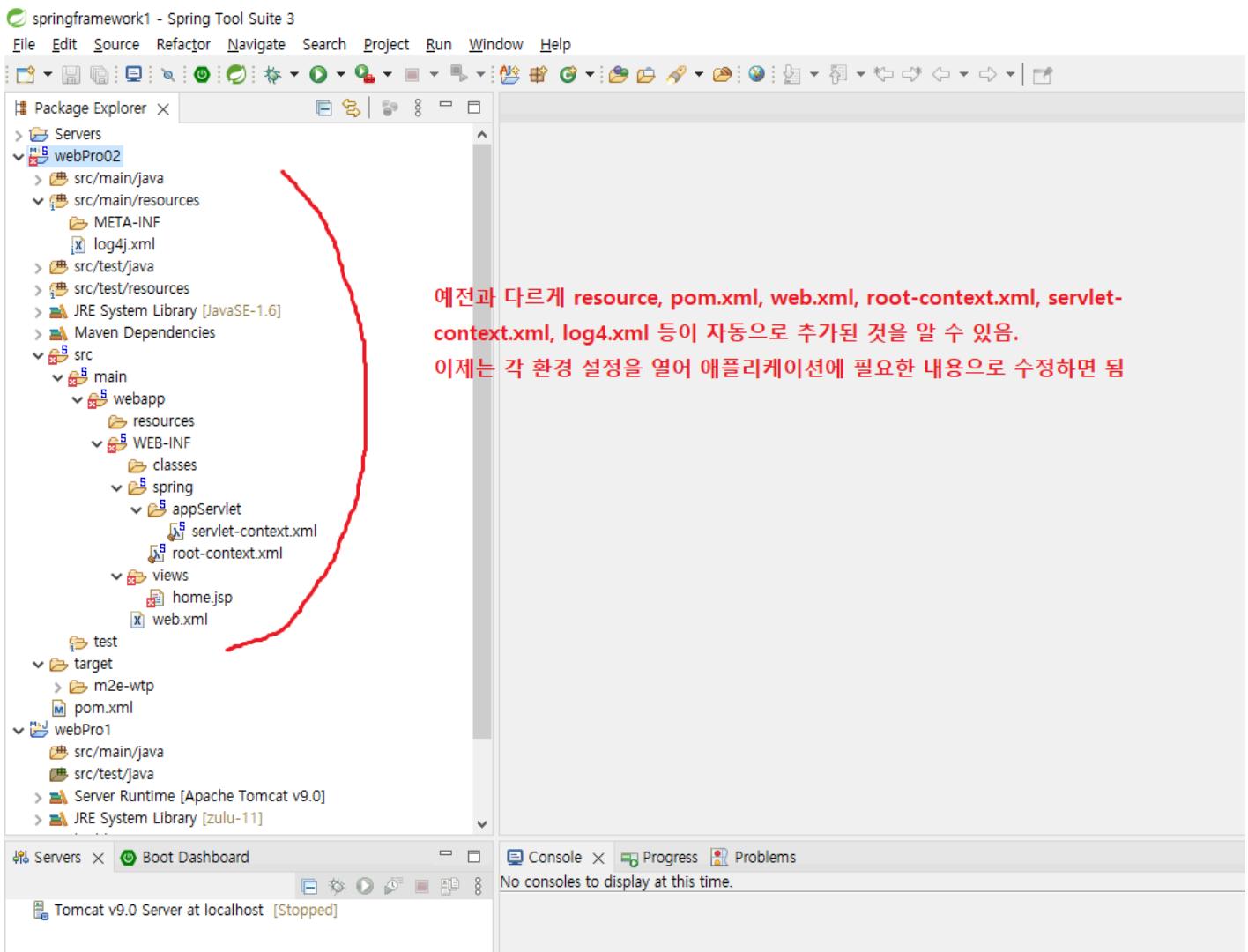
(4) 프로젝트의 이름을 확인하고, Spring MVC Project를 선택합니다.



(5) 메인 패키지의 이름을 입력합니다.

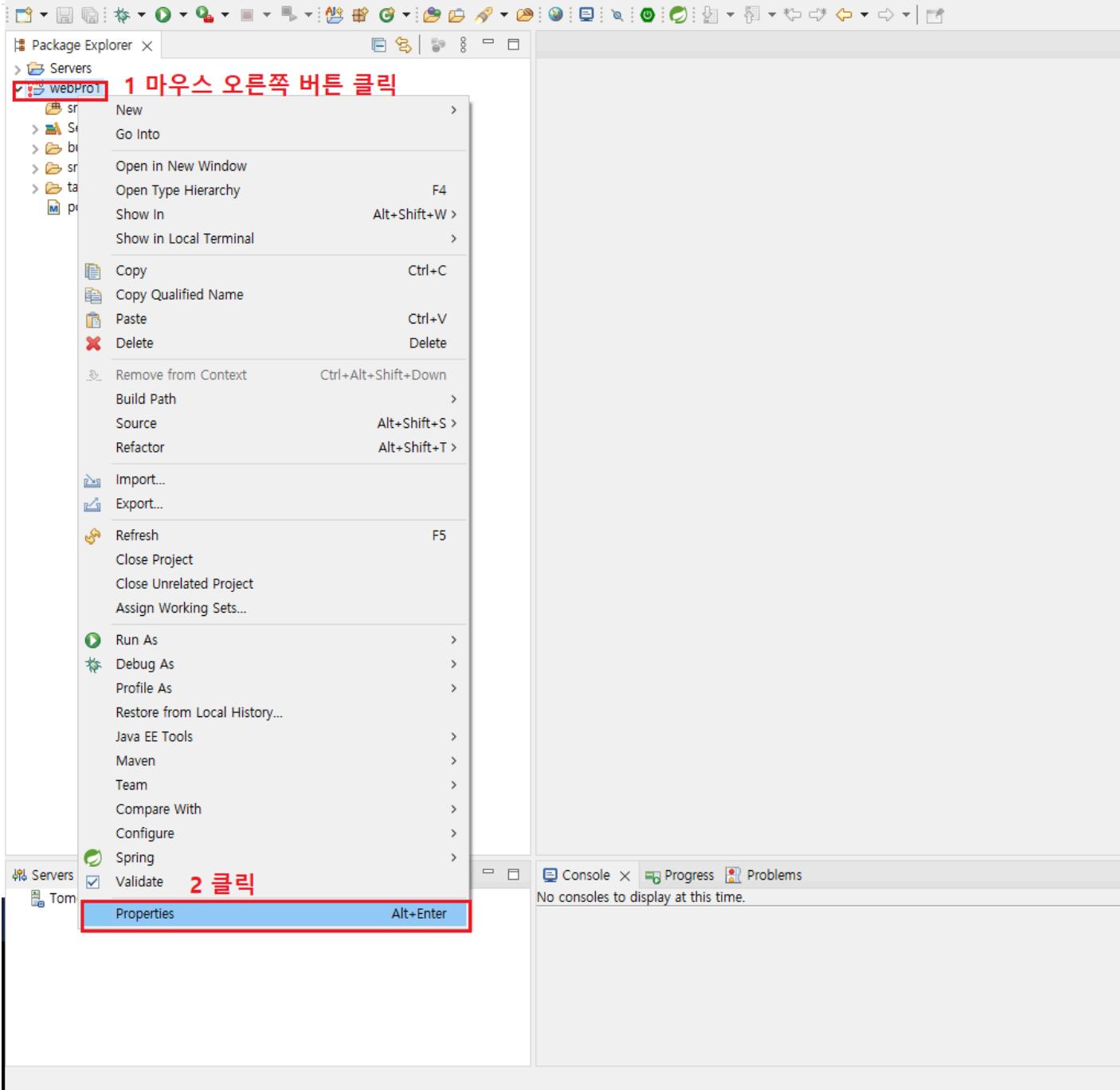


(6) 추가된 프로젝트의 내용을 상세하게 확인합니다.

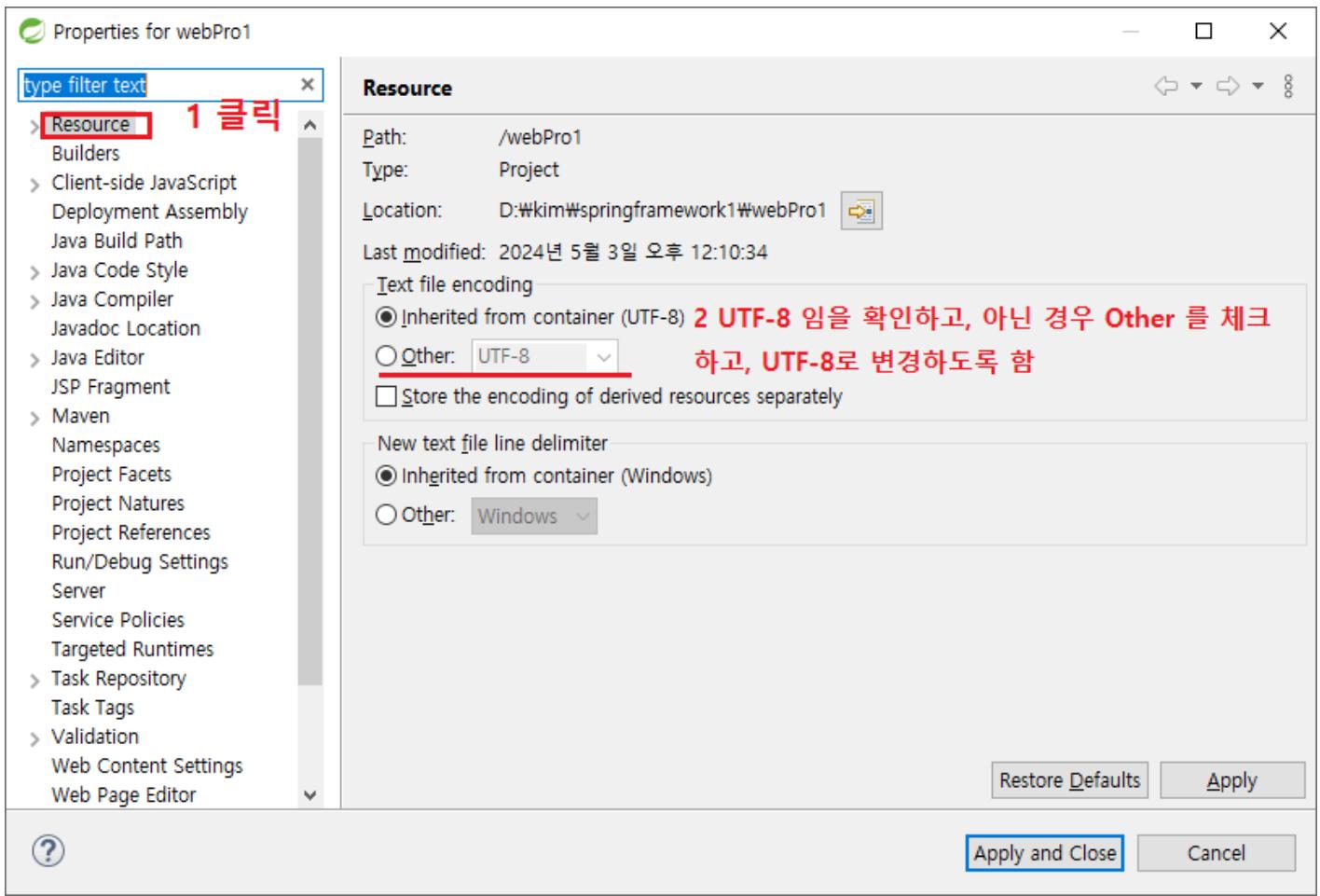


## 1-8. Spring Legacy Project 환경 설정하기

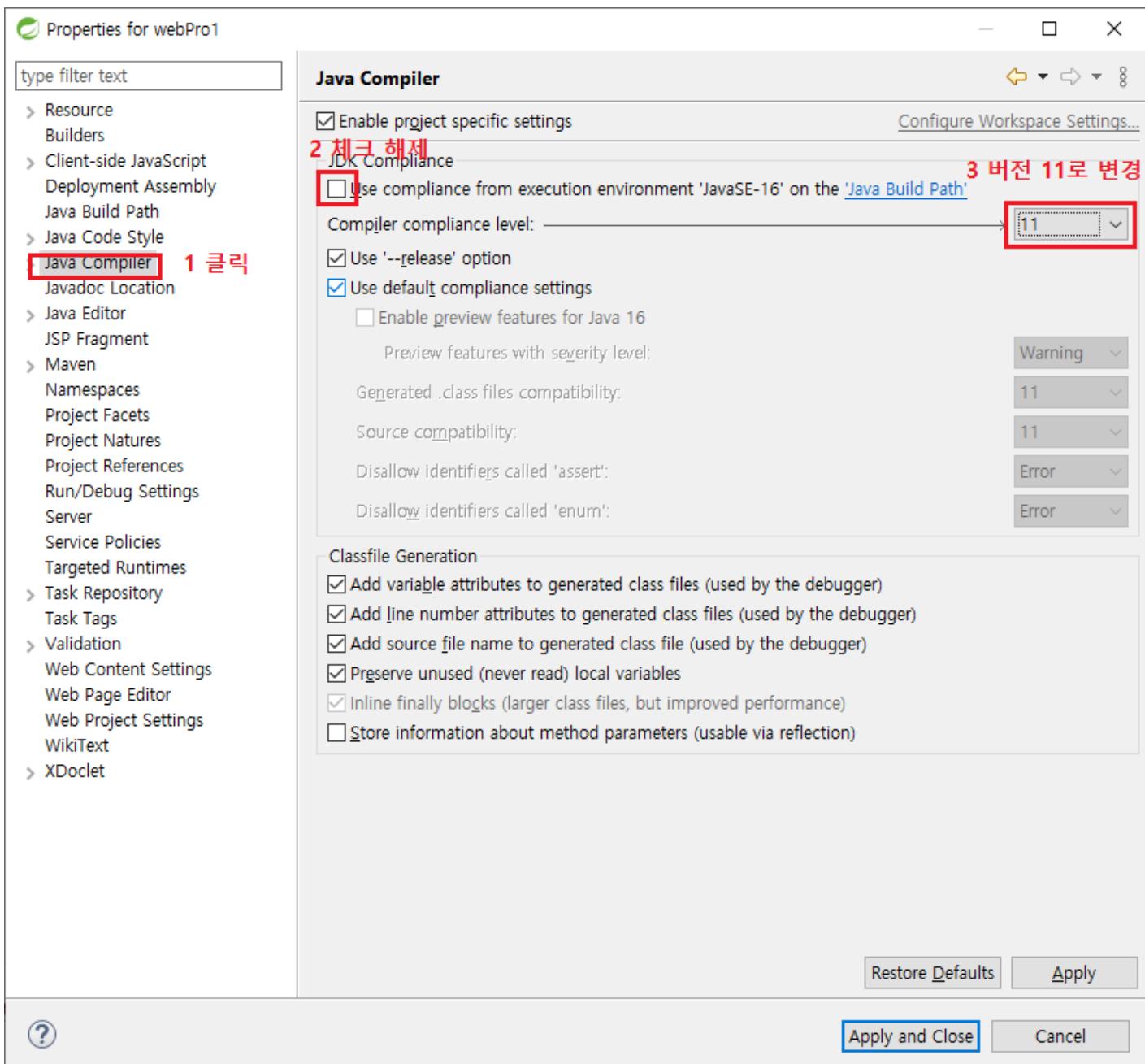
(1) 프로젝트 환경을 설정창을 불러옵니다.



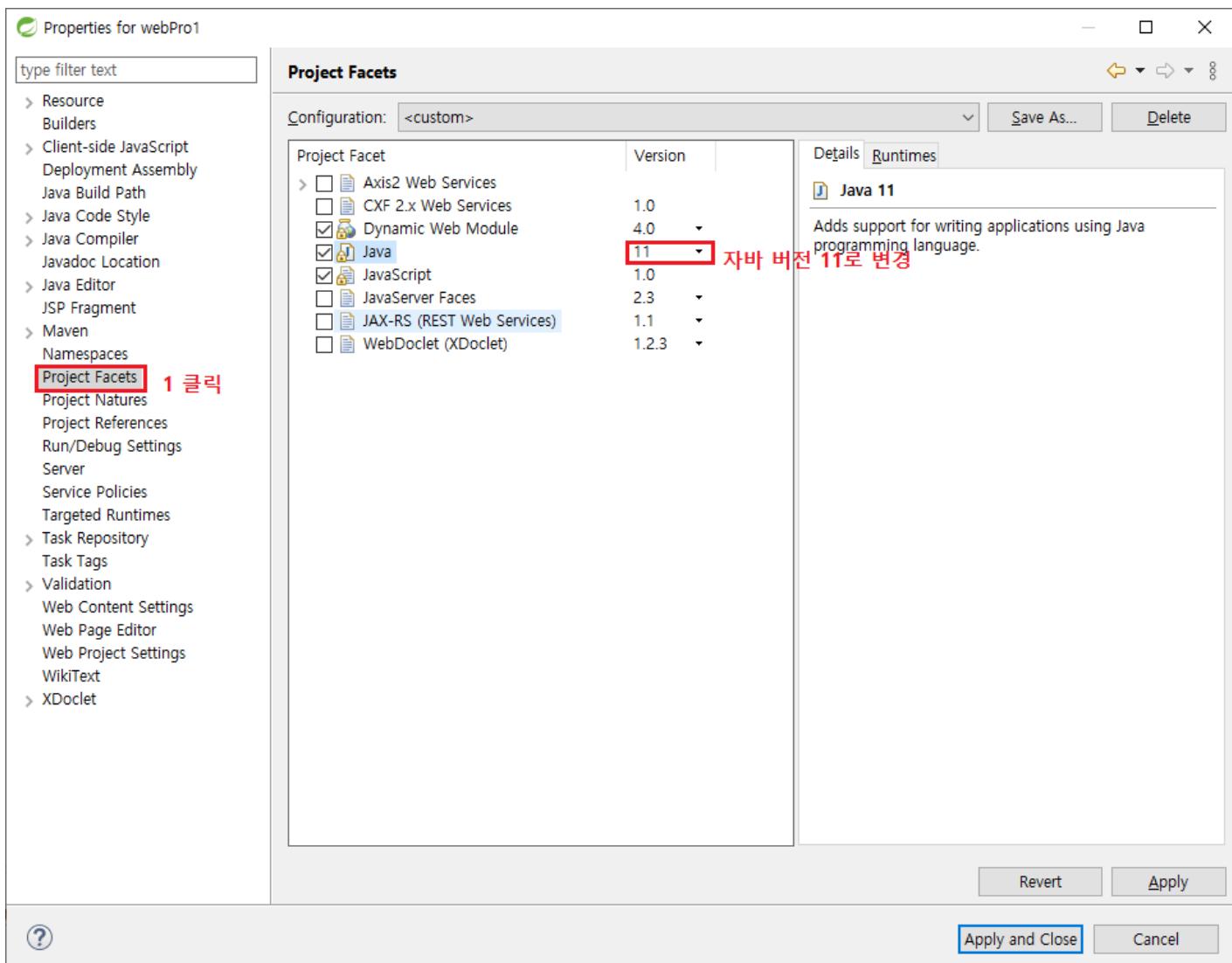
(2) 프로젝트의 인코딩 방식을 지정합니다.



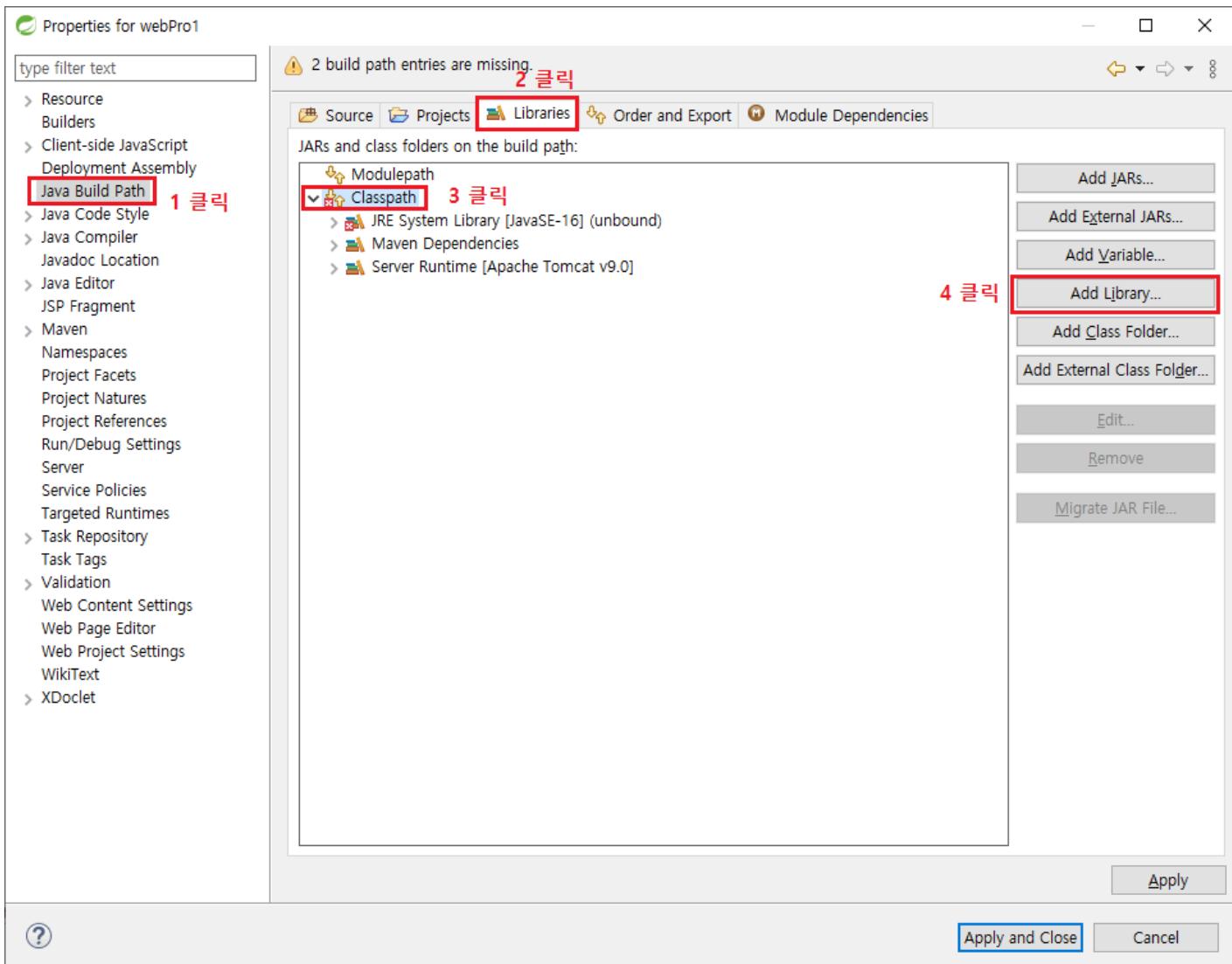
(3) 프로젝트에서 사용할 자바 컴파일러를 설정합니다.



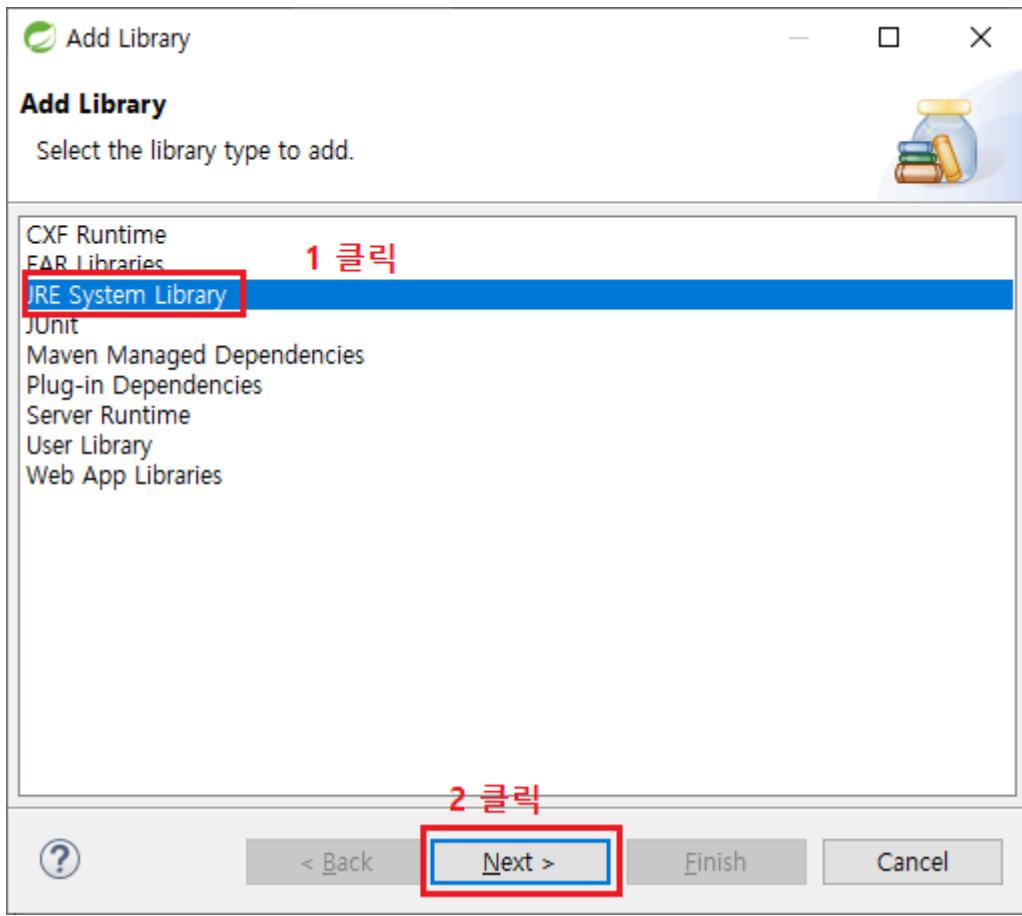
(4) 프로젝트의 프론트엔드/백엔드 웹 환경을 설정합니다.



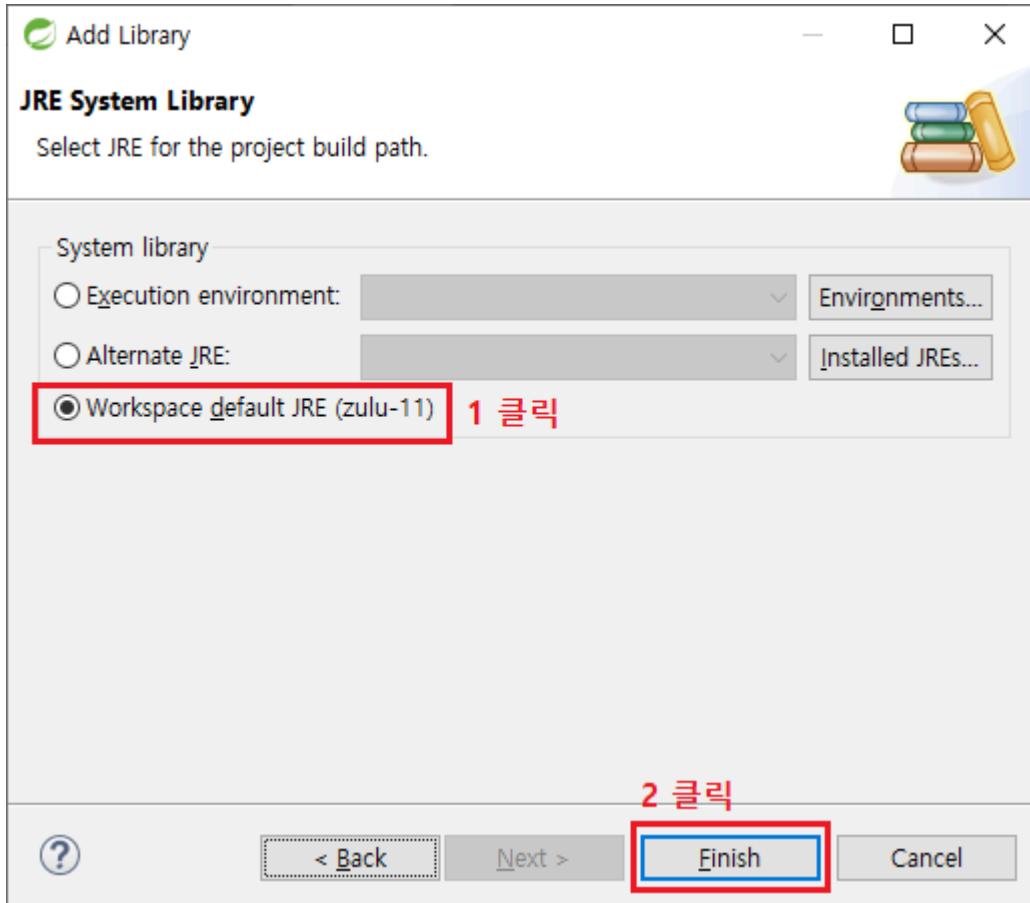
(5) 프로젝트에서 사용할 JRE 환경을 지정합니다.



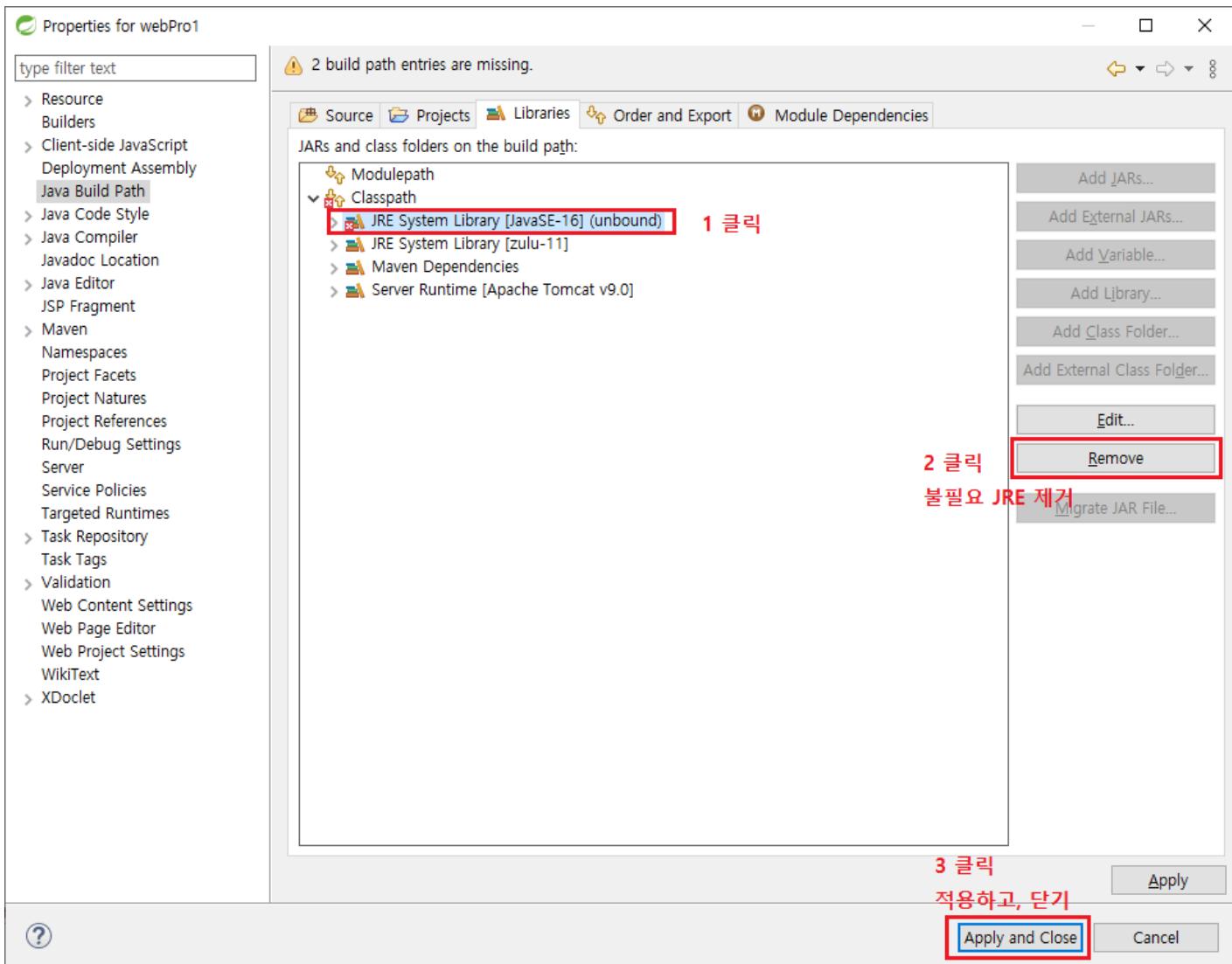
(6) 프로젝트에서 사용할 JRE 환경을 불러옵니다.



(7) 프로젝트에서 사용할 JRE 환경을 선택합니다.



(8) 프로젝트에서 사용하지 않는 JRE 환경을 제거하고, 환경 설정을 종료합니다.



## 2. 애플리케이션 개발

### 2-1. 애플리케이션 설정하기

#### 2-1-1. 프로젝트 관리자 역할을 하는 POM.xml 의존성 등록하기

아래 화면과 같이 pom.xml 파일을 열고 작성은 완료하도록 합니다.

springframework - spring1/pom.xml - Spring Tool Suite 3

File Edit Source Refactor Search Project Run Window Help

Package Explorer X

spring1/pom.xml X

1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
2     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"  
3     modelVersion="4.0.0"/>  
4 <groupId>spring1</groupId>  
5 <artifactId>spring1</artifactId>  
6 <version>0.0.1-SNAPSHOT</version>  
7 <packaging>war</packaging>  
8 <properties>  
9     java-version>11</java-version>  
10    org.springframework-version>5.0.8.RELEASE</org.springframework-version>  
11    org.aspectj-version>1.8.10</org.aspectj-version>  
12    org.slf4j-version>1.7.25</org.slf4j-version>  
13 </properties>  
14 <dependencies>  
15     <dependency>  
16         <groupId>org.springframework</groupId>  
17         <artifactId>spring-context</artifactId>  
18         <version>\${org.springframework-version}</version>  
19         <exclusions>  
20             <!-- Exclude Commons Logging in favor of SLF4j -->  
21             <exclusion>  
22                 <groupId>commons-logging</groupId>  
23                 <artifactId>commons-logging</artifactId>  
24             </exclusion>  
25         </exclusions>  
26     </dependency>  
27     <dependency>  
28         <groupId>org.springframework</groupId>  
29         <artifactId>spring-webmvc</artifactId>  
30         <version>\${org.springframework-version}</version>  
31     </dependency>  
32  
33     <!-- AspectJ : 관점지향형(AOP) 기능 제공 라이브러리 -->  
34     <dependency>  
35         <groupId>org.aspectj</groupId>  
36         <artifactId>aspectjrt</artifactId>  
37         <version>\${org.aspectj-version}</version>

Servers X Boot Dashboard

Tomcat v9.0 Server at localhost [Stopped]

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Console X Progress Problems

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>spring1</groupId>
    <artifactId>spring1</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>
    <properties>
        <java-version>11</java-version>
        <org.springframework-version>5.0.8.RELEASE</org.springframework-version>
        <org.aspectj-version>1.8.10</org.aspectj-version>
        <org.slf4j-version>1.7.25</org.slf4j-version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${org.springframework-version}</version>
            <exclusions>
                <!-- SLF4j를 위해 Commons Logging을 제외시킴 -->
                <exclusion>
                    <groupId>commons-logging</groupId>
                    <artifactId>commons-logging</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>${org.springframework-version}</version>
        </dependency>

        <!-- AspectJ : 관점지향형(AOP) 기능 제공 라이브러리 -->
        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjrt</artifactId>
            <version>${org.aspectj-version}</version>
        </dependency>

        <!-- Logging : 모든 자원의 접속 로그를 기록하는 라이브러리 -->
        <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>

<!-- @Inject : 의존성 주입 라이브러리 -->
<dependency>
    <groupId>javax.inject</groupId>
    <artifactId>javax.inject</artifactId>
    <version>1</version>
</dependency>

<!-- JSP/Servlet 라이브러리 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>

```

```
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>

<!-- Test : junit 테스트 라이브러리 -->
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.13</version>
    <scope>test</scope>
</dependency>

<!-- 스프링 테스트 라이브러리 추가 -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>5.0.8.RELEASE</version>
</dependency>

<!-- war 배포 및 패키징 라이브러리 추가 -->
<dependency>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.2.0</version>
</dependency>

<!-- getter, setter, constructer를 자동 생성해주는 라이브러리 -->
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.22</version>
    <scope>provided</scope>
</dependency>

<!-- log4jdbc-log4j2-jdbc4 : DB 접속로그를 기록하는 라이브러리 -->
<dependency>
    <groupId>org.bgee.log4jdbc-log4j2</groupId>
    <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
    <version>1.16</version>
</dependency>

<!-- 스프링 트랜잭션 라이브러리 -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>${org.springframework-version}</version>
</dependency>

<!-- 스프링 jdbc 라이브러리 -->
<dependency>
    <groupId>org.springframework</groupId>
```

```
<artifactId>spring-jdbc</artifactId>
<version>${org.springframework-version}</version>
</dependency>

<!-- commons-dbc : 자바 웹 DB 연결 공용 라이브러리 -->
<dependency>
    <groupId>commons-dbcp</groupId>
    <artifactId>commons-dbcp</artifactId>
    <version>1.4</version>
</dependency>

<!-- 오라클 jdbc 라이브러리 -->
<dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc11</artifactId>
    <version>21.1.0.0</version>
</dependency>

<!-- MySQL jdbc 라이브러리 -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.31</version>
</dependency> -->

<!-- MariaDB jdbc 라이브러리 -->
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>3.1.0</version>
</dependency> -->

<!-- SQL 구문을 XML로 쉽게 구현하기 위한 MyBatis 라이브러리 -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.4.0</version>
</dependency>

<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>1.3.2</version>
</dependency>

<!-- 구글 JSON -->
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.7</version>
</dependency>
<dependency>
```

```
<groupId>org.jsoup</groupId>
<artifactId>jsoup</artifactId>
<version>1.12.1</version>
</dependency>
<dependency>
    <groupId>org.json</groupId>
    <artifactId>json</artifactId>
    <version>20200518</version>
</dependency>

<!-- jackson 라이브러리 -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.4</version>
</dependency>

<dependency>
    <groupId>org.codehaus.jackson</groupId>
    <artifactId>jackson-mapper-asl</artifactId>
    <version>1.9.13</version>
</dependency>

<!-- 파일 첨부 및 업로드 라이브러리 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.2</version>
</dependency>
<dependency>
    <groupId>commons-io</groupId>
    <artifactId>commons-io</artifactId>
    <version>2.4</version>
</dependency>
<!-- 이미지 편집 라이브러리 -->
<dependency>
    <groupId>org.imgscalr</groupId>
    <artifactId>imgscalr-lib</artifactId>
    <version>4.0</version>
</dependency>

<!-- 자바 이메일 기본 라이브러리 -->
<dependency>
    <groupId>javax.mail</groupId>
    <artifactId>javax.mail-api</artifactId>
    <version>1.4.7</version>
</dependency>

<!-- 이메일 및 자원에 대한 외부 송출 라이브러리 -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
```

```
        <version>${org.springframework-version}</version>
    </dependency>

    <!-- java의 validation 라이브러리 -->
    <dependency>
        <groupId>javax.validation</groupId>
        <artifactId>validation-api</artifactId>
        <version>2.0.1.Final</version>
    </dependency>
    <!-- 품 검증을 애노테이션으로 검증하도록 하는 hibernate 라이브러리 -->
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-annotations</artifactId>
        <version>3.5.6-Final</version>
    </dependency>
    <!-- hibernate Validator 라이브러리 -->
    <dependency>
        <groupId>org.hibernate.validator</groupId>
        <artifactId>hibernate-validator</artifactId>
        <version>6.0.8.Final</version>
    </dependency>
    <dependency>
        <groupId>javax.xml.bind</groupId>
        <artifactId>jaxb-api</artifactId>
        <version>2.3.0</version>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <artifactId>maven-eclipse-plugin</artifactId>
            <version>2.9</version>
            <configuration>
                <additionalProjectnatures>
                    <projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>
                </additionalProjectnatures>
                <additionalBuildcommands>
                    <buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
                </additionalBuildcommands>
                <downloadSources>true</downloadSources>
                <downloadJavadocs>true</downloadJavadocs>
            </configuration>
        </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>
            </configuration>
        </plugin>
    </plugins>

```

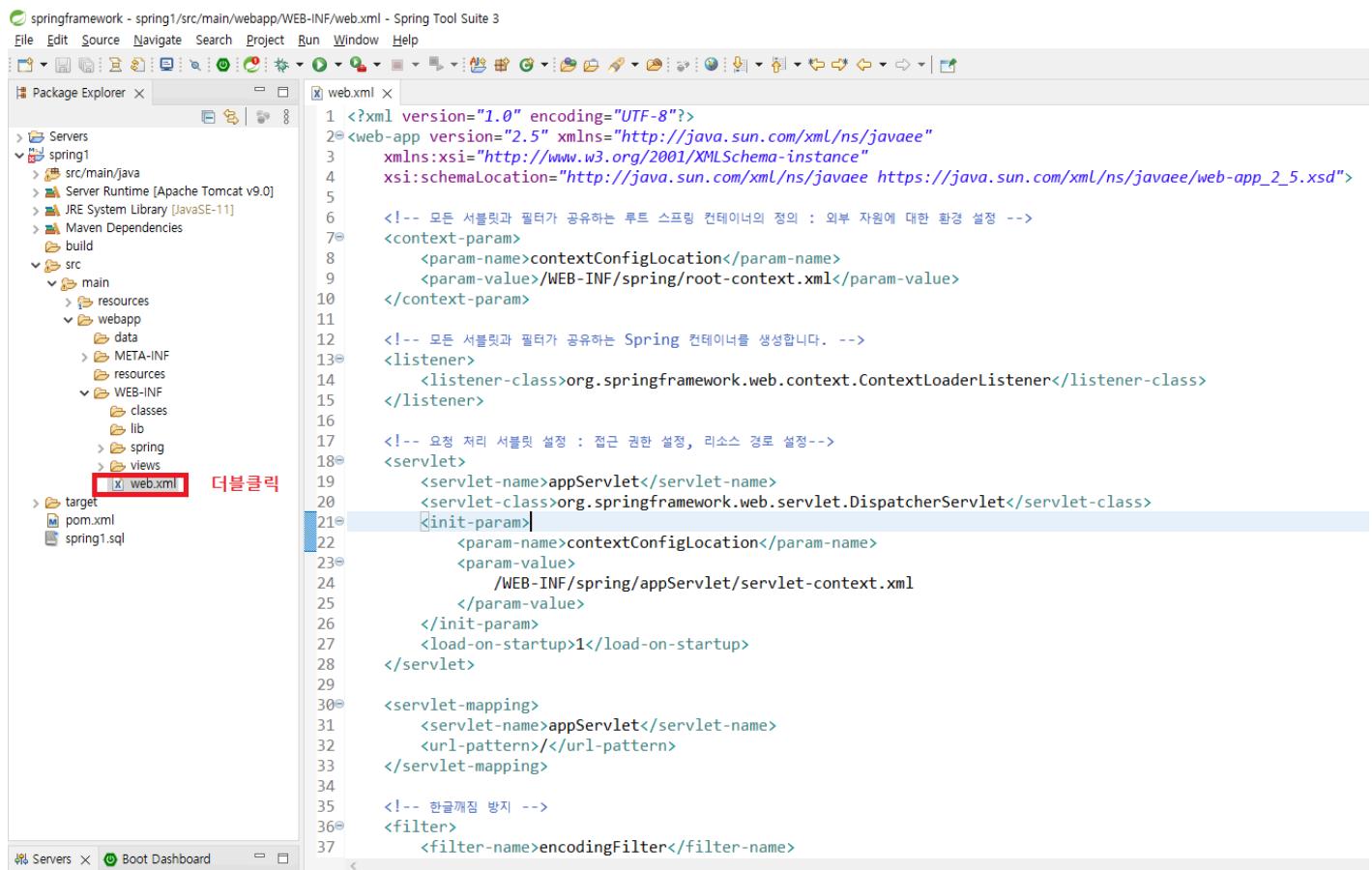
```

        <showDeprecation>true</showDeprecation>
    </configuration>
</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>
</plugins>
</build>
</project>

```

## 2-1-2. 웹 환경설정하기 - web.xml

프로젝트 이름\src\main\webapp\WEB-INF\web.xml 파일을 열고, 웹 컨테이너와 필터, 서블릿 요청 처리, 외부 자원 등을 어떻게 할지 전반적인 웹 환경을 설정합니다.



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee https://java.sun.com/xml

    <!-- 모든 서블릿과 필터가 공유하는 루트 스프링 컨테이너의 정의 : 외부 자원에 대한
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
</context-param>

    <!-- 모든 서블릿과 필터가 공유하는 Spring 컨테이너를 생성합니다. -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

    <!-- 요청 처리 서블릿 설정 : 접근 권한 설정, 리소스 경로 설정-->
<servlet>
    <servlet-name>appServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>
            /WEB-INF/spring/appServlet/servlet-context.xml
        </param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>

    <servlet-mapping>
        <servlet-name>appServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <!-- 한글깨짐 방지 -->
<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>
</filter>
<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>
```



## 2-1-3. 외부 자원 환경설정하기 - root-context.xml

프로젝트 이름\src\main\webapp\WEB-INF\spring\root-context.xml 파일을 열고, 외부 자원인 데이터베이스, 트랜잭션, 네트워크 등의 환경을 설정합니다.

The screenshot shows the Eclipse Spring Tool Suite interface. On the left, the Package Explorer shows the project structure with a red box highlighting the 'root-context.xml' file under 'src/main/webapp/WEB-INF/spring'. The main window displays the XML code for 'root-context.xml'. A red box also highlights the 'root-context.xml' file in the breadcrumb navigation bar at the bottom. The code includes definitions for a dataSource bean (DriverManagerDataSource) and a sqlSessionFactory bean (SqlSessionFactoryBean), both referencing 'mybatis-spring-1.3.2.jar'. The code is annotated with Korean comments explaining the configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:jdbc="http://www.springframework.org/schema/jdbc"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd
                           http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
                           http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
                           http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
                           http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">

    <!-- Root Context: defines shared resources visible to all other web components -->
    <!-- 데이터베이스 설정 -->
    <!-- spring-jdbc-5.0.8.RELEASE.jar 안의 드라이버매니저 연결 -->
    <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <!-- 데이터 소스 및 드라이버 설정 : log4jdbc-log4j2-jdbc4-1.16.jar -->
        <property name="driverClassName" value="net.sf.log4jdbc.sql.jdbcapi.DriverSpy"/>
        <!-- 연결 url, 사용자 아이디, 비밀번호 설정 -->
        <property name="url" value="jdbc:log4jdbc:oracle:thin:@localhost:1521:xe" />
        <property name="username" value="system" />
        <property name="password" value="1234"/>
    </bean>
    <!-- sql을 대신할 my-batis 설정 : mybatis-spring-1.3.2.jar의 세션팩토리빈클래스 연결 -->
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource" />
        <!-- mybatis 설정파일 등록-->
        <property name="configLocation" value="classpath:/mybatis-config.xml"/>
        <!-- sqlSession 객체 주입 -->
        <property name="mapperLocations" value="classpath:mappers/**/*Mapper.xml"/>
    </bean>
    <!-- sqlSession 객체 주입 -->
    <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate" destroy-method="clearCache">
        <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"/>
    </bean>
</beans>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.spri
        http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/myb
        http://www.springframework.org/schema/beans https://www.springframewor
        http://www.springframework.org/schema/context http://www.springframewo
        http://www.springframework.org/schema/aop http://www.springframework.o
        http://www.springframework.org/schema/tx http://www.springframework.or

    <!-- Root Context: defines shared resources visible to all other web component
    <!-- 데이터베이스 설정 -->
    <!-- spring-jdbc-5.0.8.RELEASE.jar 안의 드라이버매니저 연결 -->
    <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManager
    <!-- 데이터 소스 및 드라이버 설정 : log4jdbc-log4j2-jdbc4-1.16.jar -->
        <property name="driverClassName" value="net.sf.log4jdbc.sql.jdbcapi.Dr
    <!-- 연결 url, 사용자 아이디, 비밀번호 설정 -->
        <property name="url" value="jdbc:log4jdbc:oracle:thin:@localhost:1521:
            <property name="username" value="system" />
            <property name="password" value="1234"></property>
    </bean>
    <!-- sql을 대신할 my-batis 설정 : mybatis-spring-1.3.2.jar의 세션팩토리빈클래스
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource" />
        <!-- mybatis 설정파일 등록-->
        <property name="configLocation" value="classpath:/mybatis-config.xml">
            <!-- sql처럼 데이터베이스와 자바 클래스를 데이터 연관을 지어줄 파일 위치와
            <property name="mapperLocations" value="classpath:mappers/**/*Mapper.x
    </bean>
    <!-- SqlSession 객체 주입 -->
    <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate" destroy-me
        <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></co
    </bean>

    <!-- 트랜잭션 및 DB 패키지 방안 및 각종 로깅과 보안 설정 -->
    <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataS
        <property name="dataSource" ref="dataSource" />
    </bean>

    <!-- @Transactional 어노테이션 처리 -->
    <tx:annotation-driven transaction-manager="transactionManager" />

    <!-- naver/damum/google 메일 서버 설정 -->

```

## 2-1-4. 리소스 및 뷰 리졸버와 기본 패키지 설정하기 - servlet-context.xml

프로젝트 이름\src\main\webapp\WEB-INF\spring\appServlet\servlet-context.xml 파일을 열고, 외부 리소스의 접근 권한을 설정합니다.

The screenshot shows the Spring Tool Suite 3 interface. In the left sidebar, the 'Package Explorer' shows the project structure. A red box highlights the 'servlet-context.xml' file under 'appServlet'. The right side is a code editor with the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/mvc"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:beans="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd
                           http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd">

    <!-- DispatcherServlet 의 요청 처리 인프라를 정의 -->
    <!-- Spring MVC @Controller 프로그래밍 모델을 활성화합니다. -->
    <annotation-driven />

    <!-- 접근 자원에 대한 권한 설정 -->
    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resource -->
    <resources mapping="/resources/**" location="/resources/" />
    <resources mapping="/data/**" location="/data/" />

    <resources mapping="/include/**" location="/WEB-INF/views/include" />
    <resources mapping="/board/**" location="/WEB-INF/views/board" />
    <resources mapping="/member/**" location="/WEB-INF/views/member" />
    <resources mapping="/reservate/**" location="/WEB-INF/views/reservate" />
    <resources mapping="/qna/**" location="/WEB-INF/views/qna" />
    <resources mapping="/database/**" location="/WEB-INF/views/database" />
    <resources mapping="/util/**" location="/WEB-INF/views/util" />
    <resources mapping="/sample/**" location="/WEB-INF/views/sample" />
    <resources mapping="/check/**" location="/WEB-INF/views/check" />
    <resources mapping="/user/**" location="/WEB-INF/views/user" />
    <resources mapping="/free/**" location="/WEB-INF/views/free" />

    <!-- 리졸버에 대한 접두사와 접미사 설정 -->
    <!-- @Controller가 렌더링하기 위해 선택한 뷰를 /WEB-INF/views 디렉터리의 .jsp 리소스로 확인합니다. -->
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

```

The code editor has several tabs at the bottom: Source, Namespaces, Overview, beans, context, mvc. The 'Source' tab is selected. The status bar at the bottom says 'No consoles to display at this time'.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc https://www.springframew
        http://www.springframework.org/schema/beans https://www.springframewor
        http://www.springframework.org/schema/context https://www.springframew

    <!-- DispatcherServlet 의 요청 처리 인프라를 정의 -->

    <!-- Spring MVC @Controller 프로그래밍 모델을 활성화합니다. -->
<annotation-driven />

    <!-- 접근 자원에 대한 권한 설정 -->
    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up sta
<resources mapping="/resources/**" location="/resources/" />
<resources mapping="/data/**" location="/data/" />

    <resources mapping="/include/**" location="/WEB-INF/views/include" />
    <resources mapping="/board/**" location="/WEB-INF/views/board" />
    <resources mapping="/member/**" location="/WEB-INF/views/member" />
    <resources mapping="/reservate/**" location="/WEB-INF/views/reservate" />
    <resources mapping="/qna/**" location="/WEB-INF/views/qna" />
    <resources mapping="/databank/**" location="/WEB-INF/views/databank" />
    <resources mapping="/util/**" location="/WEB-INF/views/util" />
    <resources mapping="/sample/**" location="/WEB-INF/views/sample" />
    <resources mapping="/check/**" location="/WEB-INF/views/check" />
    <resources mapping="/user/**" location="/WEB-INF/views/user" />
    <resources mapping="/free/**" location="/WEB-INF/views/free" />

    <!-- 리졸버에 대한 접두사와 접미사 설정 -->
    <!-- @Controller가 렌더링하기 위해 선택한 뷰를 /WEB-INF/views 디렉터리의 .jsp 리:
<beans:bean class="org.springframework.web.servlet.view.InternalResourceViewRe
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>

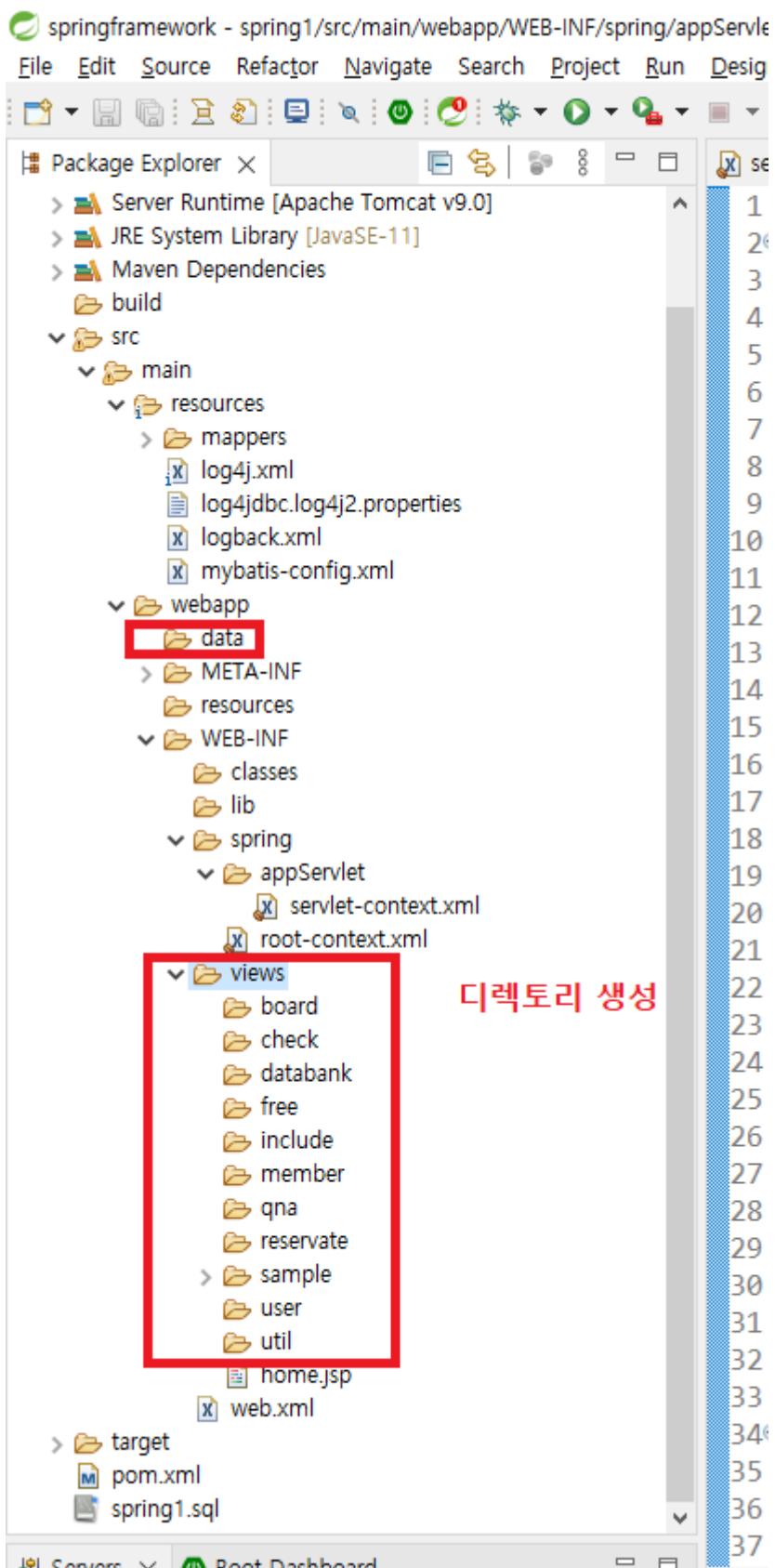
    <!-- 멀티파트 리졸버 설정 -->
<beans:bean id="multipartResolver" class="org.springframework.web.multipart.co
    <beans:property name="maxUploadSize" value="10485760" /> <!-- 10mb 제한
</beans:bean>

    <!-- 멀티파트 업로드 디렉토리 지정 -->
<beans:bean id="uploadPath" class="java.lang.String">
    <beans:constructor-arg value="D:\kim\springframework\spring1\src\main\
</beans:bean>

    <!-- 기본 메인 패키지 및 컨트롤러 패키지 설정 -->
<context:component-scan base-package="com.spring1" />
```

```
</beans:beans>
```

## 리소스 디렉토리 생성하기



## 생성할 디렉토리 이름 목록



```
프로젝트이름\src\main\webapp\data  
프로젝트이름\src\main\webapp\WEB-INF\views\include  
프로젝트이름\src\main\webapp\WEB-INF\views\board  
프로젝트이름\src\main\webapp\WEB-INF\views\member  
프로젝트이름\src\main\webapp\WEB-INF\views\reservate  
프로젝트이름\src\main\webapp\WEB-INF\views\qna  
프로젝트이름\src\main\webapp\WEB-INF\views\databank  
프로젝트이름\src\main\webapp\WEB-INF\views\util  
프로젝트이름\src\main\webapp\WEB-INF\views\sample  
프로젝트이름\src\main\webapp\WEB-INF\views\check  
프로젝트이름\src\main\webapp\WEB-INF\views\user  
프로젝트이름\src\main\webapp\WEB-INF\views\free
```

## 2-1-5. SqlMapper 마이바티스 설정하기 - mybatis-config.xml

프로젝트이름\src\main\resources\mybatis-config.xml 파일을 새로 생성하고, Mapper에 대한 내용을 설정하도록 합니다.

The screenshot shows the Spring Tool Suite 3 interface. On the left, the Package Explorer view displays the project structure. In the 'src' directory, under 'main/resources', there is a file named 'mybatis-config.xml'. This file is highlighted with a red border, and the text '더블클릭' (Double-click) is overlaid on it. To the right, the code editor window shows the XML configuration for MyBatis:

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE configuration PUBLIC  
"-//mybatis.org//DTD Config 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-config.dtd">  
<configuration>  
    <typeAliases>  
        <package name="com.spring1" />  
    </typeAliases>  
</configuration>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration PUBLIC
"-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <typeAliases>
        <package name="com.spring1" />
    </typeAliases>
</configuration>

```

## 2-1-6. logger 설정하기 - log4j.xml, log4jdbc.log4j2.properties, logback.xml

프로젝트 이름\src\main\resources\log4j.xml을 열고 어떤 객체에 대한 로그를 기록할지 설정하도록 합니다.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/Log4j/">
    <!-- Appenders -->
    <appender name="console" class="org.apache.Log4j.ConsoleAppender">
        <param name="Target" value="System.out" />
        <layout class="org.apache.Log4j.PatternLayout">
            <param name="ConversionPattern" value="%-5p: %c - %m%n" />
        </layout>
    </appender>
    <!-- level의 지정가능한 value:FATAL<ERROR<WARN<INFO<DEBUG<TRACE -->
    <!-- FATAL : 심각한 시스템 이상 내용 표시 -->
    <!-- ERROR : 요청에 대한 문제 발생시 표시 -->
    <!-- WARN : 처리는 가능하지만 경고성 메시지 발신 표시 -->
    <!-- INFO : 정보성 메시지 표시 -->
    <!-- DEBUG : 실행 내용에 대한 설명을 표시 -->
    <!-- TRACE : 실행 내용이나 추적할 경로 등을 표시 -->
    <!-- Application Loggers -->
    <logger name="com.spring1.myapp">
        <level value="info" />
    </logger>
    <logger name="com.spring1.controller">
        <level value="info" />
    </logger>
    <logger name="com.spring1.dao">
        <level value="info" />
    </logger>
    <logger name="com.spring1.dto">
        <level value="info" />
    </logger>
    <logger name="com.spring1.service">
        <level value="info" />
    </logger>

```

The screenshot shows the Spring Tool Suite interface with the 'log4j.xml' file open in the central editor. The file content is displayed above. The 'Source' tab is selected at the bottom of the editor. On the left, the 'Package Explorer' shows the project structure under 'src/main/resources'. A red box highlights the 'log4j.xml' file in the package tree. The bottom status bar shows 'Tomcat v9.0 Server at localhost [Stopped]'. The top menu bar includes 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Design', 'Window', and 'Help'.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

    <!-- Appenders -->
    <appender name="console" class="org.apache.log4j.ConsoleAppender">
        <param name="Target" value="System.out" />
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%-5p: %c - %m%n" />
        </layout>
    </appender>
    <!-- level의 지정 가능한 value:FATAL<ERROR<WARN<INFO<DEBUG<TRACE -->
    <!-- FATAL : 심각한 시스템 이상 내용 표시 -->
    <!-- ERROR : 요청에 대한 문제 발생시 표시 -->
    <!-- WARN : 처리는 가능하지만 경고성 메시지 발신 표시 -->
    <!-- INFO : 정보성 메시지 표시 -->
    <!-- DEBUG : 실행 내용에 대한 설명을 표시 -->
    <!-- TRACE : 실행 내용이나 추적할 경로 등을 표시 -->
    <!-- Application Loggers -->
    <logger name="com.spring1.myapp">
        <level value="info" />
    </logger>

    <logger name="com.spring1.controller">
        <level value="info" />
    </logger>

    <logger name="com.spring1.dao">
        <level value="info" />
    </logger>

    <logger name="com.spring1.dto">
        <level value="info" />
    </logger>

    <logger name="com.spring1.service">
        <level value="info" />
    </logger>

    <!-- 3rdparty Loggers -->
    <logger name="org.springframework.core">
        <level value="info" />
    </logger>

    <logger name="org.springframework.beans">
        <level value="info" />
    </logger>

    <logger name="org.springframework.context">
        <level value="info" />
    </logger>
```

```

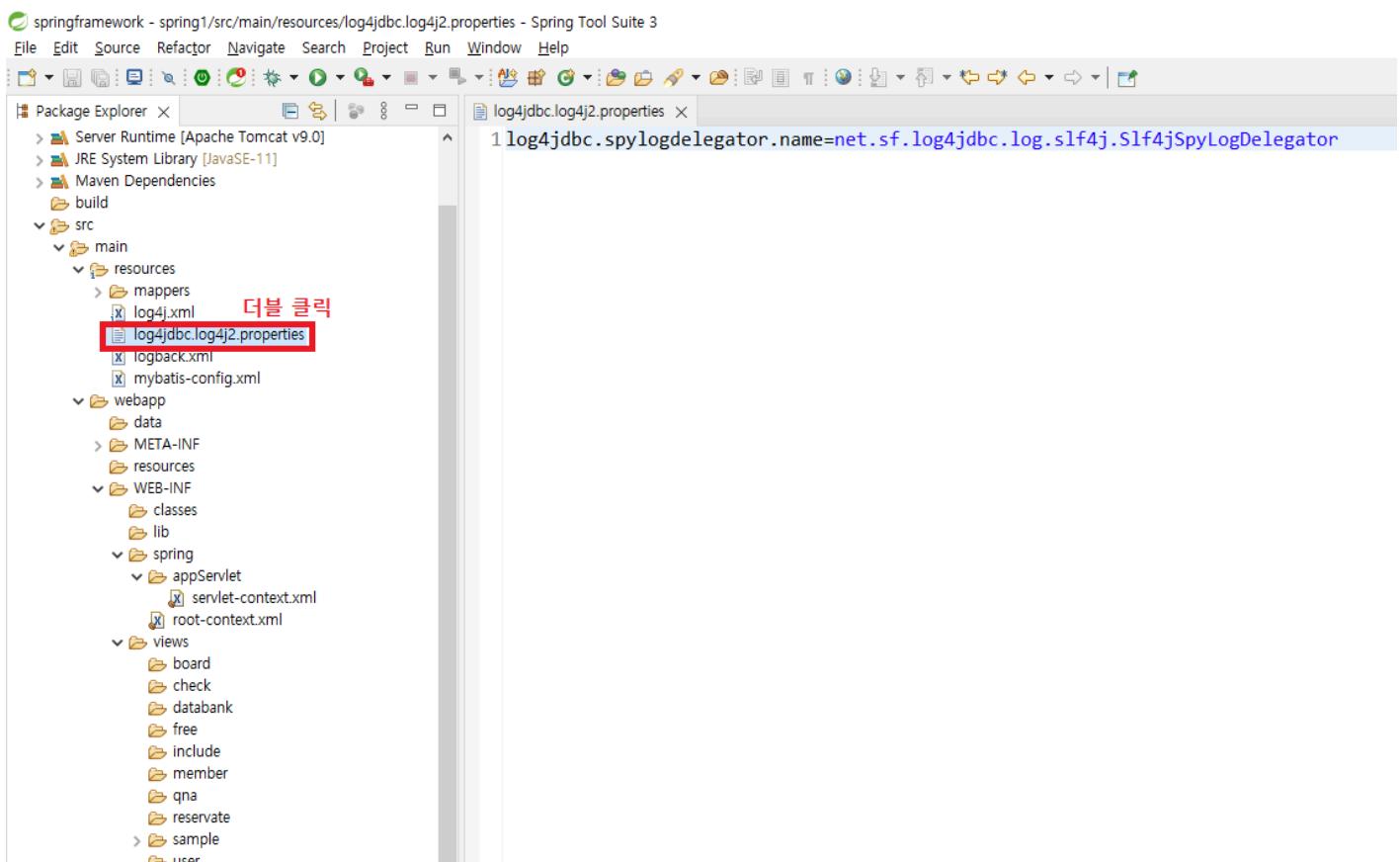
<logger name="org.springframework.web">
    <level value="info" />
</logger>

<!-- Root Logger -->
<root>
    <priority value="warn" />
    <appender-ref ref="console" />
</root>

</log4j:configuration>

```

프로젝트 이름\src\main\resources\log4jdbc.log4j2.properties 파일을 새로 만들어 로거특성정보를 저장합니다.



log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator

프로젝트 이름\src\main\resources\logback.xml 파일을 작성하여 로그정보를 주기적으로 백업할 내용을 설정합니다.

springframework - spring1/src/main/resources/logback.xml - Spring Tool Suite 3

File Edit Source Refactor Navigate Search Project Run Design Window Help

Package Explorer X logback.xml X

Server Runtime [Apache Tomcat v9.0]  
JRE System Library [JavaSE-11]  
Maven Dependencies  
build  
src  
main  
resources  
mappers  
log4j.xml  
log4jdb.log4j2.properties  
**logback.xml** 더블클릭  
mybatis-config.xml

webapp  
data  
META-INF  
resources  
WEB-INF  
classes  
lib  
spring  
appServlet  
servlet-context.xml  
root-context.xml  
views  
board  
check  
database  
free  
include  
member  
qna  
reserve  
sample  
user  
util  
home.jsp  
web.xml

target  
pom.xml  
spring1.sql

Servers X Boot Dashboard

Tomcat v9.0 Server at localhost [Stopped]

Design Source

Console X Progress Problems

No consoles to display at this time.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration>
<configuration>
    <!-- log4jdbc-log4j2 -->
    <logger name="jdbc.sqlonly" level="DEBUG"/>
    <logger name="jdbc.sqltiming" level="INFO"/>
    <logger name="jdbc.audit" level="WARN"/>
    <logger name="jdbc.resultset" level="ERROR"/>
    <logger name="jdbc.resultsettable" level="ERROR"/>
    <logger name="jdbc.connection" level="INFO"/>
    <!-- 로그의 레벨( 지정된 로그 레벨 이상만 수집 ) : DEBUG < INFO < WARN < ERROR < FATAL -->
    <logger name="myweb" additivity="false">
        <level value="INFO" />
        <appender-ref ref="LOGFILE" />
        <appender-ref ref="CONSOLE" />
    </logger>
    <appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <pattern>%d{HH:mm:ss.SSS} [%thread] %-4level [%logger.%method:%line]-%msg%</pattern>
        </layout>
    </appender>
    <appender name="LOGFILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>/WEB-INF/logback.log</file>
        <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <fileNamePattern>logback.%d{yyyy-MM-dd}.log</fileNamePattern>
            <!-- 30일 지난 파일은 삭제한다. -->
            <maxHistory>30</maxHistory>
        </rollingPolicy>
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %-4level [%logger.%method:%line]-%msg%</pattern>
        </encoder>
    </appender>
</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <!-- log4jdbc-log4j2 -->
    <logger name="jdbc.sqlonly"      level="DEBUG"/>
    <logger name="jdbc.sqltiming"    level="INFO"/>
    <logger name="jdbc.audit"        level="WARN"/>
    <logger name="jdbc.resultset"    level="ERROR"/>
    <logger name="jdbc.resultsettable" level="ERROR"/>
    <logger name="jdbc.connection"   level="INFO"/>

    <appender name="CONSOLE" class="ch.qos.logback.core.ConsoleAppender">
        <layout class="ch.qos.logback.classic.PatternLayout">
            <pattern>%d{HH:mm:ss.SSS} [%thread] %-4level [%logger.%method:
                %msg%n</pattern>
        </layout>
    </appender>

    <appender name="LOGFILE"
              class="ch.qos.logback.core.rolling.RollingFileAppender">
        <file>/WEB-INF/logback.log</file>
        <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <fileNamePattern>logback.%d{yyyy-MM-dd}.log</fileNamePattern>
            <!-- 30일 지난 파일은 삭제한다. -->
            <maxHistory>30</maxHistory>
        </rollingPolicy>
        <encoder>
            <pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} %-4level [%logger.%method
                - %msg %n</pattern>
        </encoder>
    </appender>

    <!-- 로그의 레벨( 지정된 로그 레벨 이상만 수집 ) : DEBUG < INFO < WARN < ERROR <
    <logger name="myweb" additivity="false">
        <level value="INFO" />
        <appender-ref ref="LOGFILE" />
        <appender-ref ref="CONSOLE" />
    </logger>

    <root>
        <level value="INFO" />
        <appender-ref ref="CONSOLE" />
    </root>

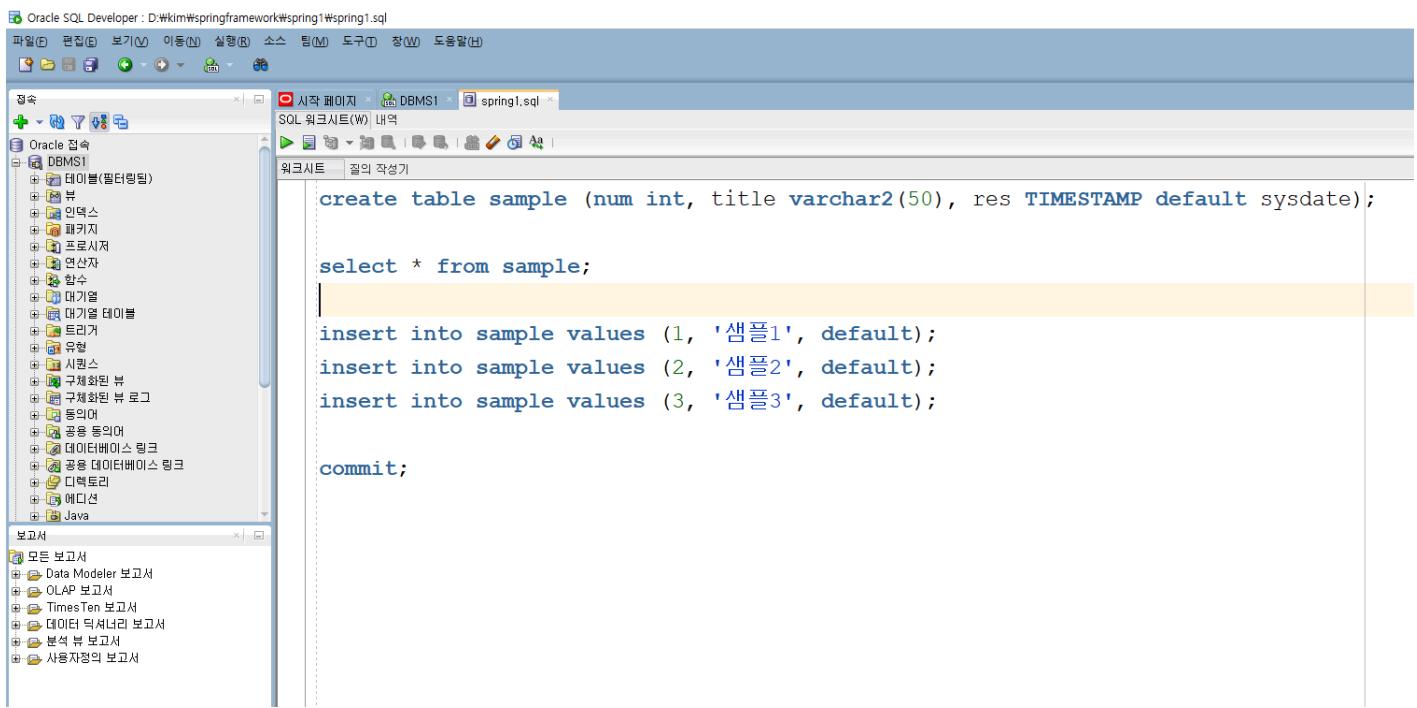
```

```
</configuration>
```

## 2-2. 영속 계층 개발작업하기

### 2-2-1. 테이블 정의와 더미 데이터 추가 작업

SQL Developer 를 실행하고, 오라클을 접속한 후 샘플(sample) 테이블을 작성합니다.



The screenshot shows the Oracle SQL Developer interface. On the left, the 'Oracle 접속' (Connection) sidebar lists various database objects like tables, indexes, triggers, and sequences. The main workspace contains the following SQL code:

```
create table sample (num int, title varchar2(50), res TIMESTAMP default sysdate);

select * from sample;

insert into sample values (1, '샘플1', default);
insert into sample values (2, '샘플2', default);
insert into sample values (3, '샘플3', default);

commit;
```

```
create table sample (num int, title varchar2(50), res TIMESTAMP default sysdate);

select * from sample;

insert into sample values (1, '샘플1', default);
insert into sample values (2, '샘플2', default);
insert into sample values (3, '샘플3', default);

commit;
```

### 2-2-2. SqlMapper(MyBatis xml file) 작성하기

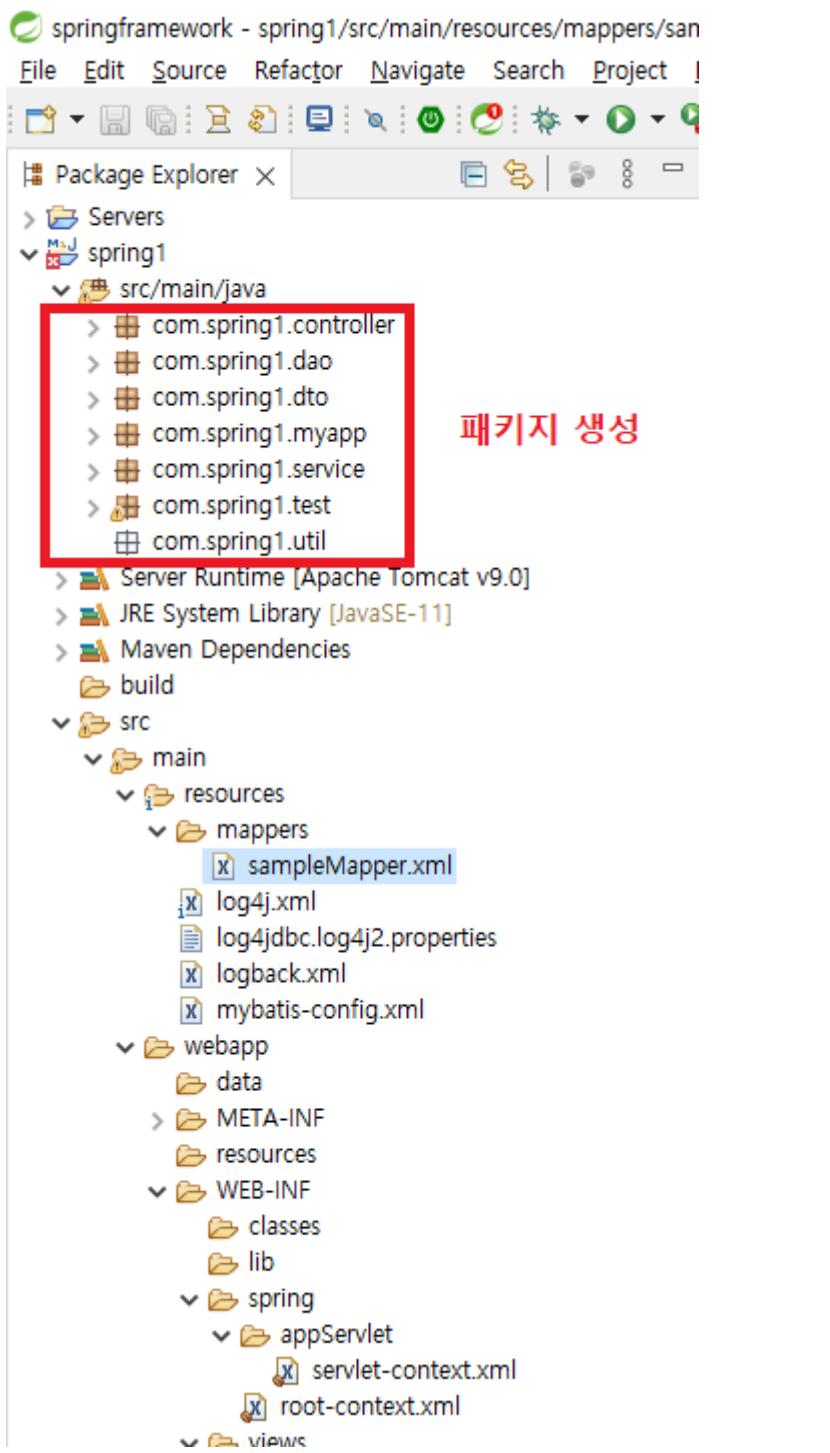
프로젝트 이름\src\main\resources\mappers\sampleMapper.xml 파일을 새로 생성하고, sample에 대한 sql 명령과 그에 해당하는 xml태그를 작성하도록 합니다.

The screenshot shows the Spring Tool Suite 3 interface. The left pane is the 'Package Explorer' showing the project structure:

- Servers
- spring1
  - src/main/java
    - Apache Tomcat v9.0
    - JRE System Library [JavaSE-11]
  - Maven Dependencies
  - build
  - src
    - main
      - resources
        - mappers
      - webapp
        - data
        - META-INF
        - resources
        - WEB-INF
          - classes
          - lib
        - spring
          - appServlet
            - dispatcher-servlet.xml
          - views
            - board

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC
"-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="sample">
    <select id="getSampleList" resultType="com.spring1.dto.Sample">
        select * from sample
    </select>
    <select id="getSample" resultType="com.spring1.dto.Sample">
        select * from sample where num=#{num}
    </select>
    <insert id="insSample">
        insert into sample values (#{num}, #{title}, #{res})
    </insert>
    <update id="upSample">
        update sample set title=#{title} where num=#{num}
    </update>
    <delete id="delSample">
        delete from sample where num=#{num}
    </delete>
</mapper>
```

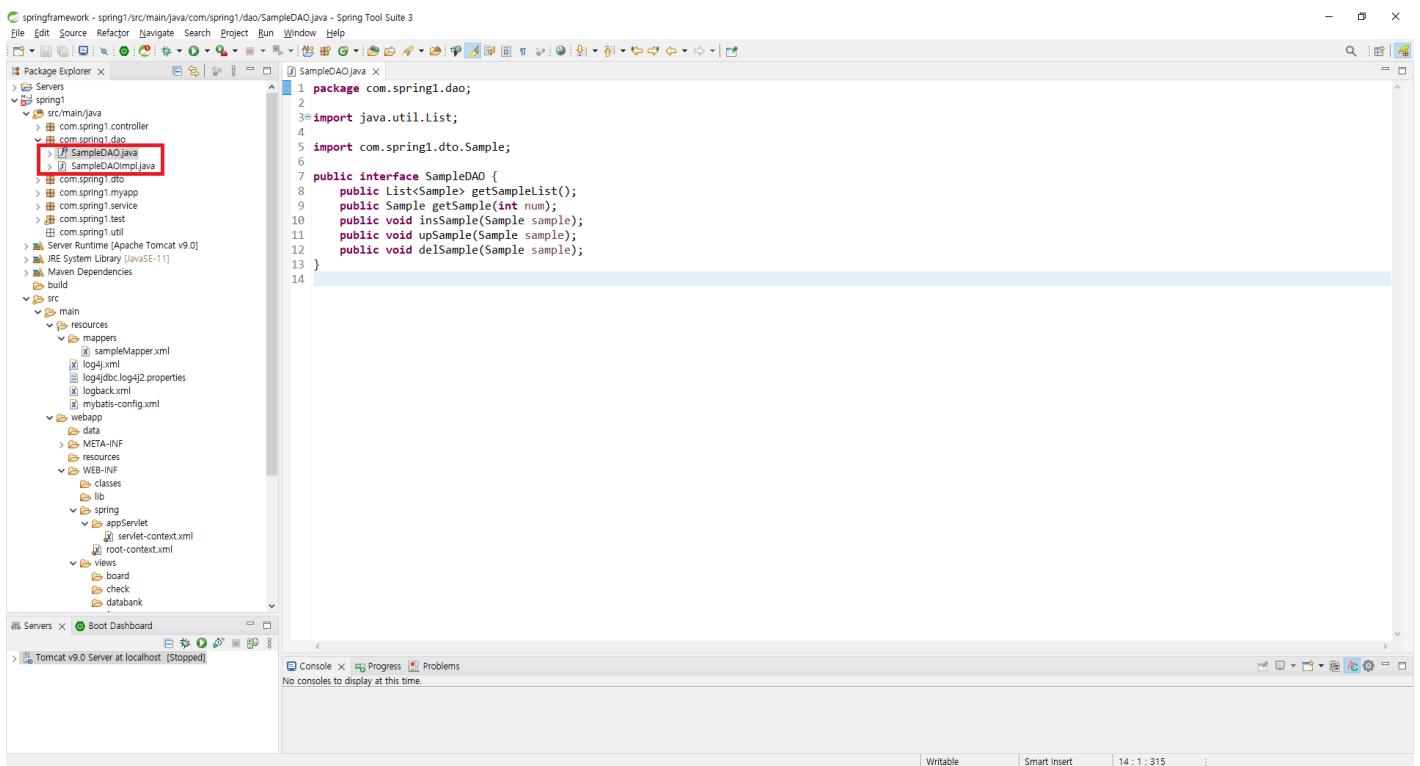
프로젝트에 작성할 각종 패키지를 생성합니다.



프로젝트이름\src\main\java\com\프로젝트이름\controller  
프로젝트이름\src\main\java\com\프로젝트이름\dao  
프로젝트이름\src\main\java\com\프로젝트이름\dto  
프로젝트이름\src\main\java\com\프로젝트이름\myapp  
프로젝트이름\src\main\java\com\프로젝트이름\service  
프로젝트이름\src\main\java\com\프로젝트이름\test  
프로젝트이름\src\main\java\com\프로젝트이름\util

## 2-2-3. Repository(DAO) 작성하기

## 프로젝트이름\src\main\java\com\프로젝트이름\dao\SampleDAO.java 작성



```
package com.spring1.dao;

import java.util.List;

import com.spring1.dto.Sample;

public interface SampleDAO {
    public List<Sample> getSampleList();
    public Sample getSample(int num);
    public void insSample(Sample sample);
    public void upSample(Sample sample);
    public void delSample(Sample sample);
}
```



## 프로젝트이름\src\main\java\com\프로젝트이름\dao\SampleDAOImpl.java 작성



```
package com.spring1.dao;

import java.util.List;

import org.apache.ibatis.session.SqlSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import com.spring1.dto.Sample;

@Repository
public class SampleDAOImpl implements SampleDAO {

    @Autowired
    private SqlSession sqlSession;

    @Override
    public List<Sample> getSampleList() {
        return sqlSession.selectList("sample.getSampleList");
    }

    @Override
    public Sample getSample(int num) {
        return sqlSession.selectOne("sample.getSample", num);
    }

    @Override
    public void insSample(Sample sample) {
        sqlSession.insert("sample.insSample", sample);
    }

    @Override
    public void upSample(Sample sample) {
        sqlSession.update("sample.upSample", sample);
    }

    @Override
    public void delSample(Sample sample) {
        sqlSession.delete("sample.delSample", sample);
    }
}
```

## 2-2-4. Service 작성하기

프로젝트이름\src\main\java\com\프로젝트이름service\SampleService.java 작성

The screenshot shows the Spring Tool Suite interface. On the left, the Package Explorer view displays the project structure under the 'spring1' project. A red box highlights the 'src/main/java/com/spring1/service' folder, which contains 'SampleService.java' and 'SampleServiceImpl.java'. The right side shows the code editor for 'SampleService.java' with the following content:

```

1 package com.spring1.service;
2
3 import java.util.List;
4
5 import com.spring1.dto.Sample;
6
7 public interface SampleService {
8     public List<Sample> getSampleList();
9     public Sample getSample(int num);
10    public void insSample(Sample sample);
11    public void upSample(Sample sample);
12    public void delSample(Sample sample);
13 }
14

```

At the bottom, the Servers view shows a stopped Tomcat v9.0 Server at localhost.

```

package com.spring1.service;

import java.util.List;

import com.spring1.dto.Sample;

public interface SampleService {
    public List<Sample> getSampleList();
    public Sample getSample(int num);
    public void insSample(Sample sample);
    public void upSample(Sample sample);
    public void delSample(Sample sample);
}

```

## 프로젝트이름\src\main\java\com\프로젝트이름\service\SampleServiceImpl.java 작성

The screenshot shows the Spring Tool Suite 3 interface. The left side features the Package Explorer with a tree view of the project structure. A red box highlights the 'SampleServiceImpl.java' file under the 'com.spring1.service' package. The right side shows the code editor with the content of 'SampleServiceImpl.java'. The code implements the 'SampleService' interface, using autowiring to inject the 'SampleDAO' dependency. It overrides methods to get, insert, update, and delete samples from the DAO.

```
1 package com.spring1.service;
2
3 import java.util.List;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Service;
6
7 import com.spring1.dao.SampleDAO;
8 import com.spring1.dto.Sample;
9
10 @Service
11 public class SampleServiceImpl implements SampleService {
12
13     @Autowired
14     private SampleDAO sampleDAO;
15
16     @Override
17     public List<Sample> getSampleList() {
18         return sampleDAO.getSampleList();
19     }
20
21     @Override
22     public Sample getSample(int num) {
23         return sampleDAO.getSample(num);
24     }
25
26     @Override
27     public void insSample(Sample sample) {
28         sampleDAO.insSample(sample);
29     }
30
31     @Override
32     public void upSample(Sample sample) {
33         sampleDAO.upSample(sample);
34     }
35
36     @Override
37     public void delSample(Sample sample) {
38         sampleDAO.delSample(sample);
39     }
40 }
```

```
package com.spring1.service;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.spring1.dao.SampleDAO;
import com.spring1.dto.Sample;

@Service
public class SampleServiceImpl implements SampleService {

    @Autowired
    private SampleDAO sampleDAO;

    @Override
    public List<Sample> getSampleList() {
        return sampleDAO.getSampleList();
    }

    @Override
    public Sample getSample(int num) {
        return sampleDAO.getSample(num);
    }

    @Override
    public void insSample(Sample sample) {
        sampleDAO.insSample(sample);
    }

    @Override
    public void upSample(Sample sample) {
        sampleDAO.upSample(sample);
    }

    @Override
    public void delSample(Sample sample) {
        sampleDAO.delSample(sample);
    }
}
```



## 2-2-5. Controller 작성하기

프로젝트이름\src\main\java\com\프로젝트이름\service\SampleController.java 작성

```
package com.spring1.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;

import com.spring1.dto.Sample;
import com.spring1.service.SampleService;

@Controller
@RequestMapping("/sample/")
public class SampleController {

    @Autowired
    private SampleService sampleService;

    @GetMapping("list.do")
    public String getSampleList(Model model) {
        List<Sample> list = sampleService.getSampleList();
        model.addAttribute("list", list);
        return "sample/sampleList";
    }
}
```

프로젝트 이름\src\main\java\com\spring1\myapp\HomeController.java 수정하기

```
package com.spring1.myapp;

import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@Controller
public class HomeController {

    private static final Logger log = LoggerFactory.getLogger(HomeController.class)
    //localhost:8091/spring1/ => /WEB-INF/views/home.jsp
    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        log.info("Welcome home~! The Client locale is {}.", locale);

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateInstance(DateFormat.LONG

        String formatDate = dateFormat.format(date);

        model.addAttribute("serverTime", formatDate);
        model.addAttribute("author", "김기태");
        model.addAttribute("company", "파람소프트");

        return "home";
    }
}
```

## 2-2-6. View(jsp) 작성하기

프로젝트이름\src\main\webapp\WEB-INF\views\home.jsp 수정하기

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="fn" uri = "http://java.sun.com/jsp/jstl/functions"%>
<c:set var="path2" value="${pageContext.request.contextPath }" />
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>메인 페이지</title>
</head>
<body>
    <h2>${serverTime }</h2>
    <h2>${author }</h2>
    <h2>${company }</h2>
    <hr>
    <a href="${path2 }/sample/list.do">샘플 목록</a>
</body>
</html>

```

## 프로젝트 이름\src\main\webapp\WEB-INF\views\sample\sampleList.jsp 작성

The screenshot shows the Spring Tool Suite interface. On the left, the Package Explorer view displays the project structure. In the center, the code editor shows the content of the sampleList.jsp file.

```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4  <%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
5  <%@ taglib prefix="fn" uri = "http://java.sun.com/jsp/jstl/functions"%>
6  <!DOCTYPE html>
7  <html>
8  <head>
9  <meta charset="UTF-8">
10 <title>샘플 목록</title>
11 </head>
12 <body>
13     <c:forEach var="sample" items="${list}">
14         <p>번호 : ${sample.num }</p>
15         <p>제목 : ${sample.title }</p>
16         <p>일시 : ${sample.res }</p>
17         <hr>
18     </c:forEach>
19 </body>
20 </html>

```

The file sampleList.jsp is located in the views\sample directory under the webapp folder. The code in the file uses JSP tags to iterate over a list of samples and display their details.

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"  %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="fn" uri = "http://java.sun.com/jsp/jstl/functions"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>샘플 목록</title>
</head>
<body>
    <c:forEach var="sample" items="${list}">
        <p>번호 : ${sample.num }</p>
        <p>제목 : ${sample.title }</p>
        <p>일시 : ${sample.res }</p>
        <hr>
    </c:forEach>
</body>
</html>
```

