

스프링 프레임워크

본자료의 무단 복제 및 전제, 배포를 금지합니다.

Spring

개발환경

Spring

Java Platform

◆ Java Platform 환경

- ◆ Spring Framework는 JDK 버전에 의존적이다.
- ◆ JDK 버전 로드맵
 - <https://www.oracle.com/java/technologies/java-se-support-roadmap.html>

JDK	서비스기간
8	LTS Public(~2019.1), Extended(~2030.3)
11	LTS (2018.9 ~ 2023.9), Extended(~2026.9)
17	LTS (2021.9 ~ 2026.9), Extended(~2029.9)
21	LTS (2023.9 ~ 2028.9), Extended(~2031.9)

- ◆ Spring MVC는 Servlet 버전에 의존적이다.
 - <https://tomcat.apache.org/whichversion.html>

JDK	Servlet	package	JSP	EL	Tomcat
8 ~	4.0	javax.*	2.3	3.0	9.0.x
11 ~	6.0	Jakarta.*	3.1	5.0	10.1.x
21 ~	6.1	-	4.0	6.0	11.0.x

Spring

Java Platform

◆ Spring

- <https://github.com/spring-projects/spring-framework/wiki/Spring-Framework-Versions>
- 스프링 버전을 기준으로 표기하였다.

JDK	eclipse	JEE	Spring Framework	SpringBoot
8	~ 4.16 (2020-06)	~ 7	4.3.x ~ 5.3.x	2.0.x ~ 2.7.x
11	4.17 (2020-09) ~	7 ~ 8	5.1.x ~ 5.3.x	2.1.x ~ 2.7.x
17	4.25 (2022-09) ~	9 ~	5.3.x ~ 6.1.x	3.0.x ~

• Spring MVC 기준

- Spring 6.x 부터는 패키지 명을 Jakarta.* 를 이용하고, 이전 버전은 javax.* 를 이용해야 한다.
- Spring 6.x 부터는 최소 Tomcat 10 이상 이용해야 한다.

JDK	Spring Framework	SpringBoot	Tomcat	Servlet	package
8 ~ 17	5.3.x	2.7.x	9.0.x	4.0	javax.*
17~21	6.1.x	3.0.x ~	10.1.x	6.0	Jakarta.*

Spring

■ 개발환경

◆ STS (Spring Tool Suite)

- ◆ 스프링 커뮤니티에서 공식적으로 지원하는 통합 개발환경이다.
- ◆ 이클립스를 기반으로 제작되어 있어 다양한 이클립스의 플러그인을 이용할 수 있다.
- ◆ 비주얼 스튜디오 코드(Visual Studio Code), 이클립스(Eclipse), 테이아(Theia) 등 IDE에서 플러그인(Plug In)으로 이용할 수 있다.
- ◆ 설치
 - Spring Boot
 - STS 3.x, 4.x 버전을 이용한다.
 - <https://spring.io/tools> 에서 다운로드 가능하다.
 - Spring Legacy
 - STS 3.x 버전을 이용한다.
 - <https://spring.io/tools> 하단의 링크를 이용하여 다운로드 가능하다.

Looking for Spring Tool Suite 3?

Version 3 of the Spring Tool Suite is longer under active development and does not receive any maintenance updates anymore. The last and final release can be found on the [Spring Tool Suite 3 wiki](#), alongside details of how to upgrade to **Spring Tools 4**.

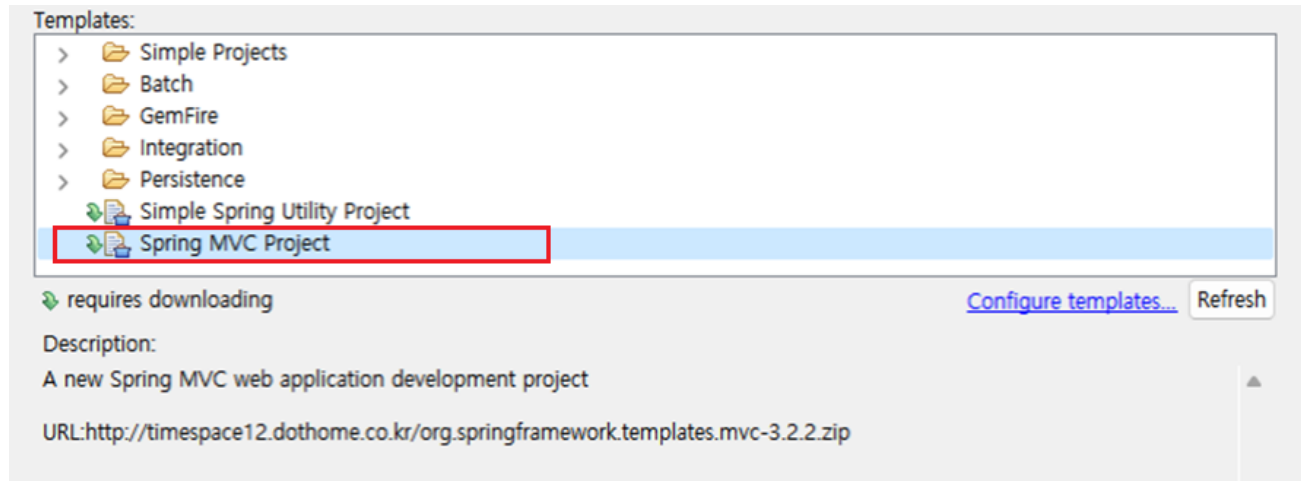
- <https://github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3>
 - JDK8 인 경우 Eclipse 4.16 기반의 3.9.15 을 권장한다.
- 설치 폴더에 압축을 푼다.

Spring

I 개발환경

◆ STS3

- "Spring Legacy Project"를 선택한 뒤 "**Spring MVC Project**" 항목이 보이지 않을 경우 다음과 같이 설정한다.

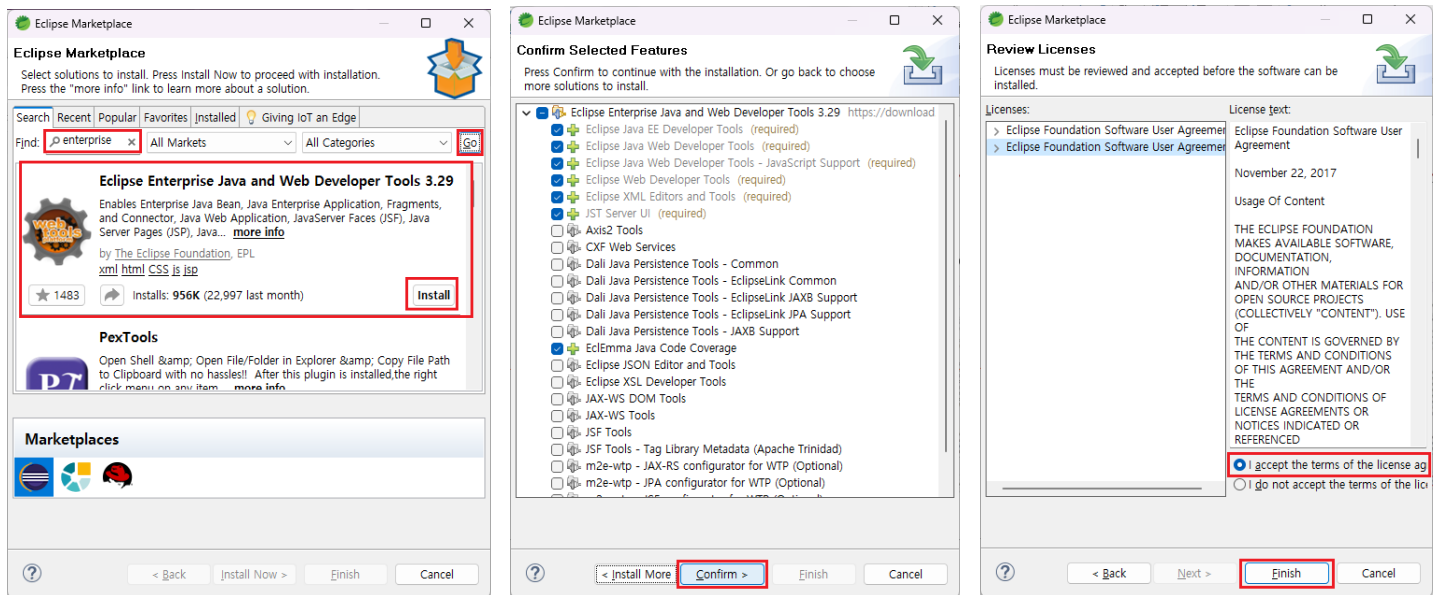


- Workspace 경로의 하위 경로에 "https-content.xml"를 저장한다.
 - %workspace%\metadata\plugins\org.springframework.ide.eclipse.commons.content.core
- STS3를 다시 실행한다.

Spring

개발환경

- ◆ STS4
 - Dynamic Web Project 활성화 (선택)
 - Help > Eclipse Marketplace...
 - Eclipse Enterprise Java and Web Developer Tools ...



Spring

개발환경

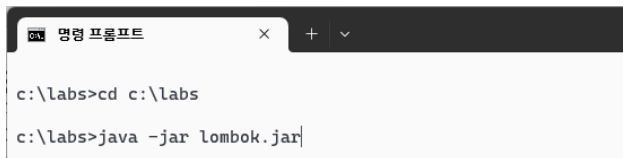
◆ Lombok

◆ 롬복(Lombok)

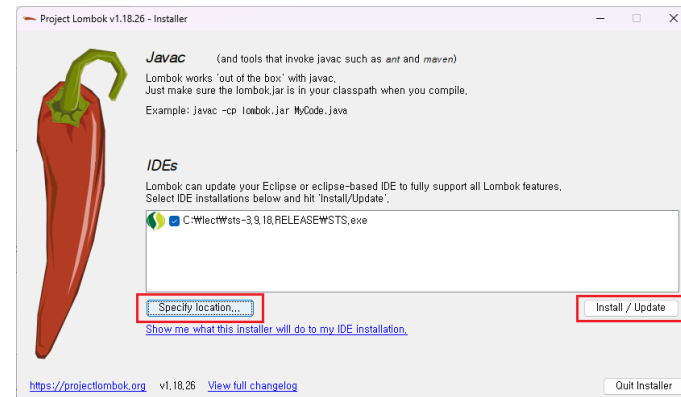
- 자바 응용 프로그램 제작 시 생산성 향상을 위해 IDE 및 빌드 도구(build tool)에 설치하는 자바 라이브러리이다.
- 생성자, Getter/Setter, Logging 등 반복적으로 사용되는 코드 패턴(Code Pattern)을 다양한 어노테이션(Annotation)을 이용하여 간편하게 구현할 수 있다.

◆ 설치

- <https://projectlombok.org/download> 에서 "lombok.jar"를 다운로드 받는다.
- 명령 프롬프트로 다운로드 받은 경로로 이동한 뒤 다음과 같이 실행한다.
- `c:\labs> java -jar lombok.jar`



```
명령 프롬프트
c:\labs>cd c:\labs
c:\labs>java -jar lombok.jar
```



- 설치가 완료되면 [Quit Installer] 버튼을 클릭한다.

개발환경

◆ 주요 어노테이션(Annotation)

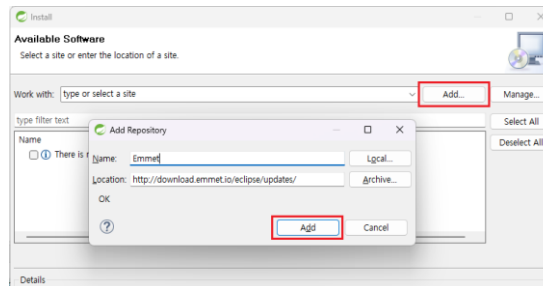
Annotation	설명	대상
@Getter, @Setter	클래스의 멤버 변수에 해당 하는 getter, setter 메서드를 생성한다.	Class, 멤버 변수
@NoArgsConstructor	기본 생성자를 생성한다.	Class
@AllArgsConstructor	모든 멤버 변수를 매개 변수로 하는 생성자를 생성한다.	Class
@RequiredArgsConstructor	final, @NonNull인 멤버 변수만 매개 변수로 하는 생성자를 생성한다.	Class
@ToString	모든 멤버 변수를 출력하도록 toString 메서드를 오버라이딩한다.	Class
@EqualsAndHashCode	equals, hashCode 를 오버라이딩한다.	Class
@Data	@ToString, @EqualsAndHashCode, @Getter, @Setter, @RequiredArgsConstructor 를 모두 적용한다.	Class
@Builder	Builder 패턴을 이용하여 객체를 생성할 수 있게 한다.	Class
@NonNull	Null을 허용하지 않는다. 롬복으로 생성된 메서드나 생성자에서 동작한다.	멤버 변수, 메서드 (매개변수)
@Slf4j, @Log4j2	지정된 로그로 static final log 객체를 생성한다.	-
@Cleanup	지정된 자원의 사용이 완료되면 자동으로 close()가 호출된다.	멤버 변수
@SneakyThrows(xxException.class)	지정된 예외처리 구현을 메서드에 자동 생성한다.	메서드

Spring

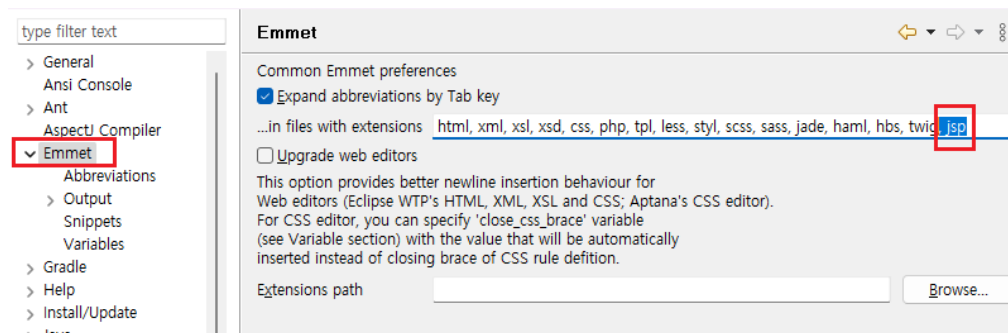
개발환경

◆ 에밋(Emmet)

- ◆ Zen-coding 이라고도 한다. 편집기에서 CSS의 선택자 문법을 이용하여 html, xml 등과 비슷한 구조의 코드들을 편리하게 생성하는 기능을 제공한다.
- ◆ 설치
 - Help > Install New Software...
 - Location : <http://download.emmet.io/eclipse/updates/>
 - "Contact all update sites during install to find required software"를 해제한다.



- ◆ Window > Preferences > Emmet



개발환경

◆ 주요 표기법

표기법	예제	결과
# .	div#list.content.link	<div id="list" class="content link"></div>
[]	div[title="abcd" alt="234"]	<div title="abcd" alt="234"></div>
> + {} *	ul>li*3+div{abc}	<div>abc</div>
() \$	(dt+dd#i\${abc\$})*2	<dt></dt><dd id="i01">abc1</dd> <dt></dt><dd id="i02">abc2</dd>

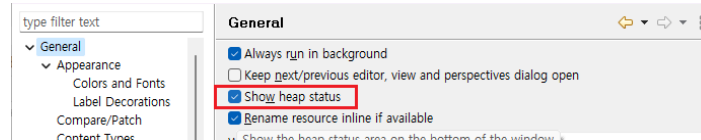
Spring

I 개발환경

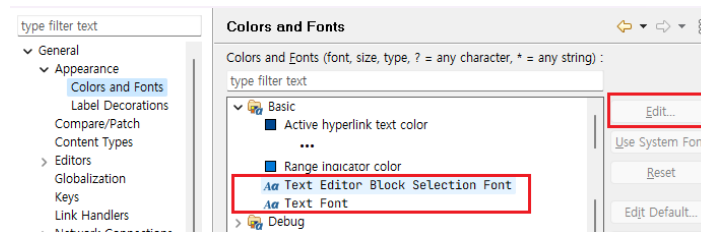
◆ IDE 환경설정

◆ Window > Preferences

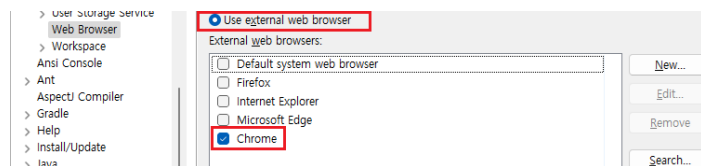
• General



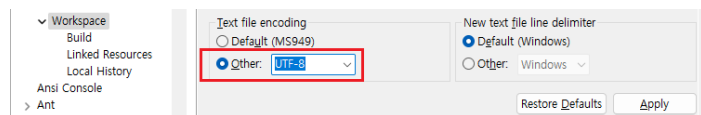
• General > Colors and Fonts



• General > Web Browser



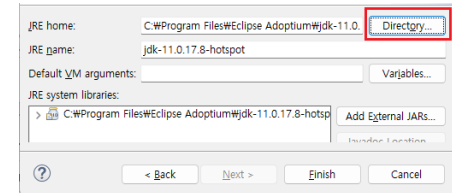
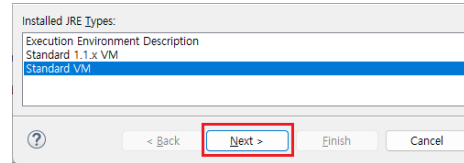
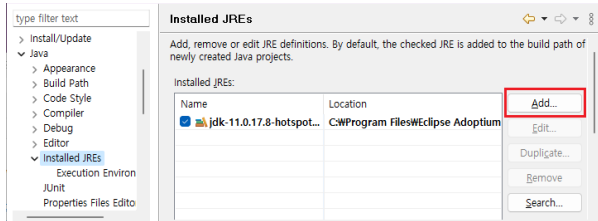
• General > Workspace



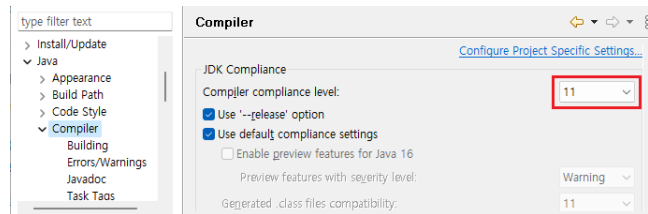
Spring

개발환경

- Java > Installed JREs



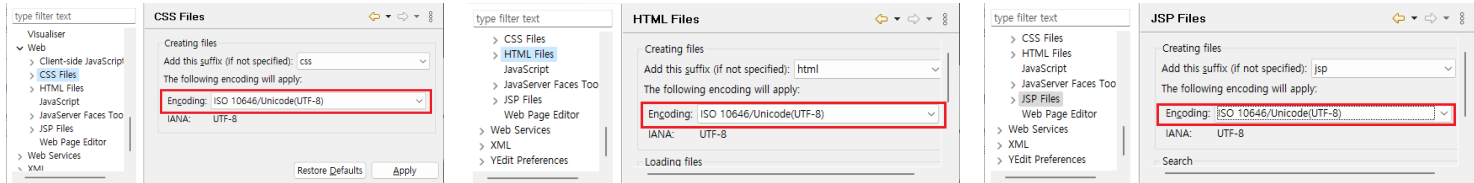
- Java > Compiler



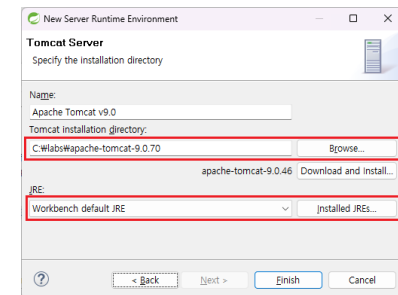
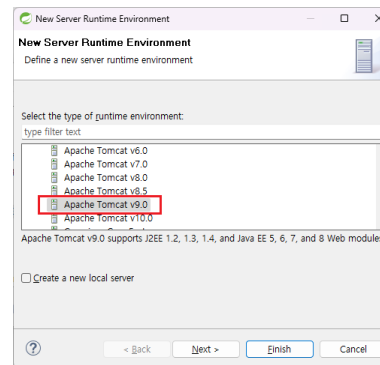
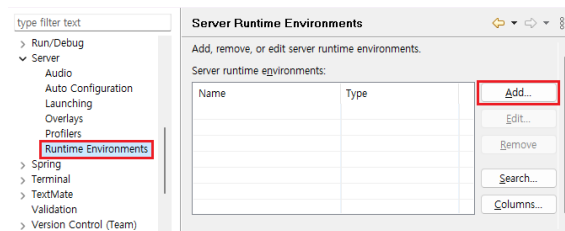
Spring

개발환경

- Web > CSS Files, HTML Files, JSP Files



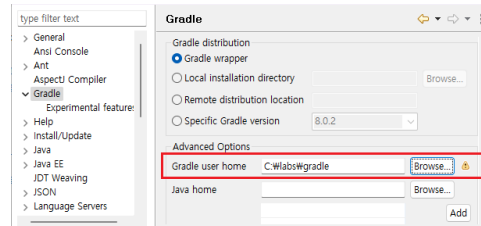
- Server > Runtime Environments



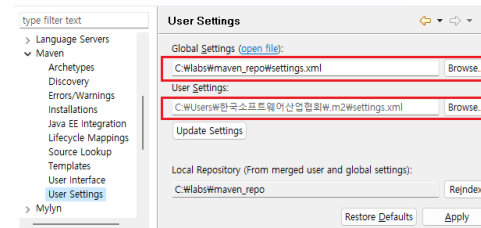
Spring

개발환경

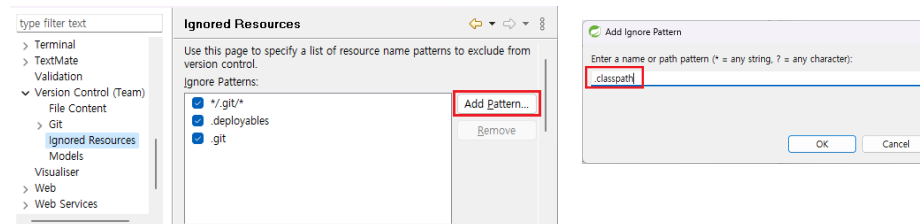
- Gradle



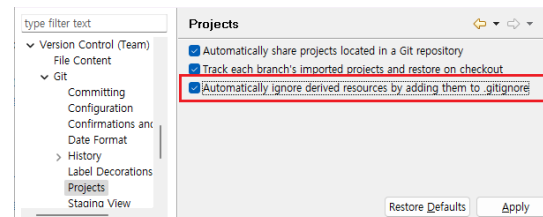
- Maven > User Settings



- Version Control (Team)



- Version Control (Team) > Git > Project



Spring

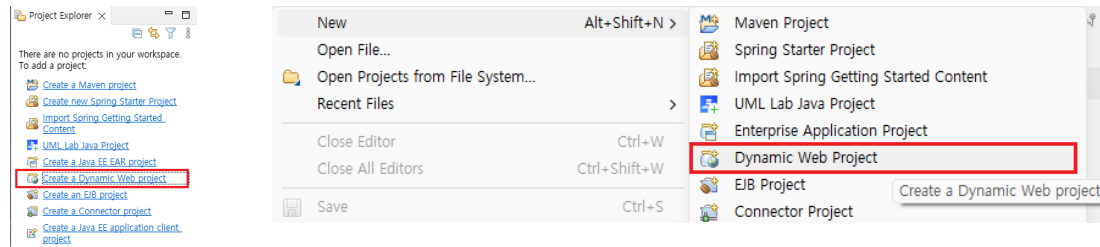
프로젝트 설정

Spring

프로젝트 설정

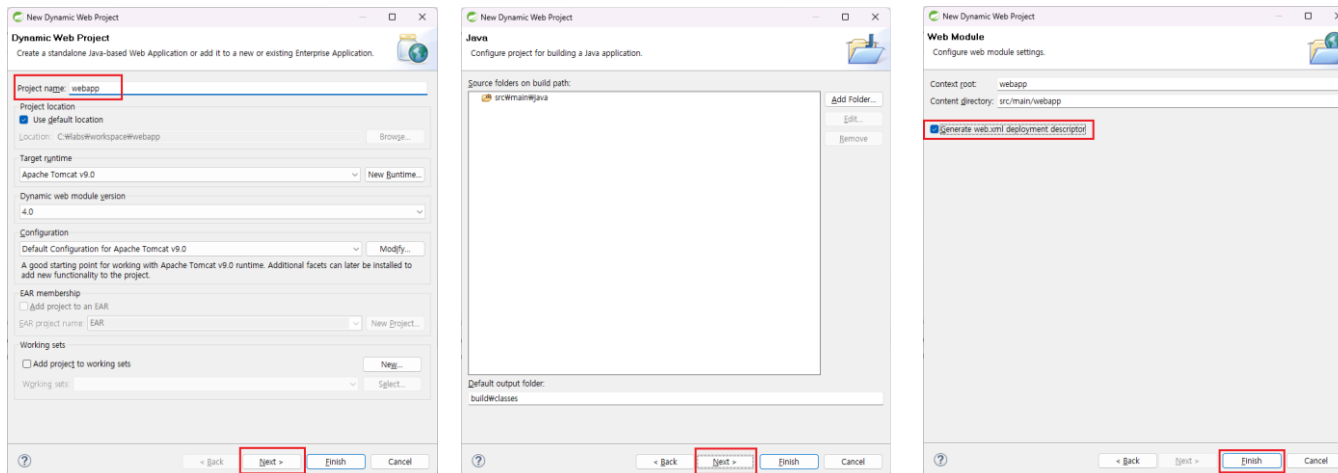
◆ Dynamic Web Project

- ◆ Project(Package) Explorer > Create Dynamic Web Project
- ◆ File > New > Dynamic Web Project



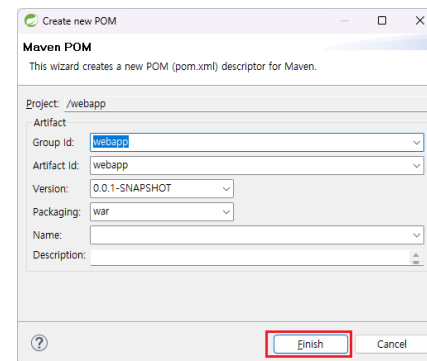
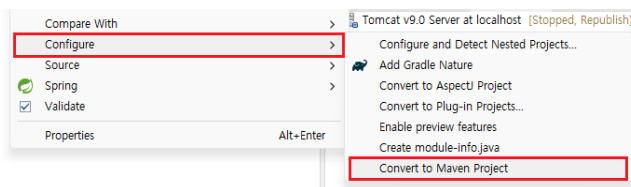
◆ 프로젝트생성

- Project Name : webapp

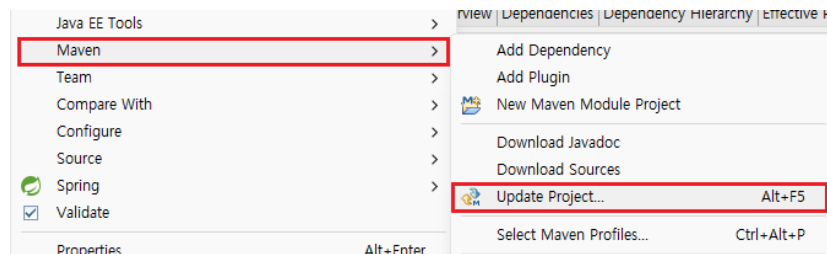


프로젝트 설정

- ◆ 공통 설정 확인을 진행한다.
- ◆ Maven 설정
 - Project > Properties, Project {M:R} > Properties
 - Configure > Convert to Maven Project



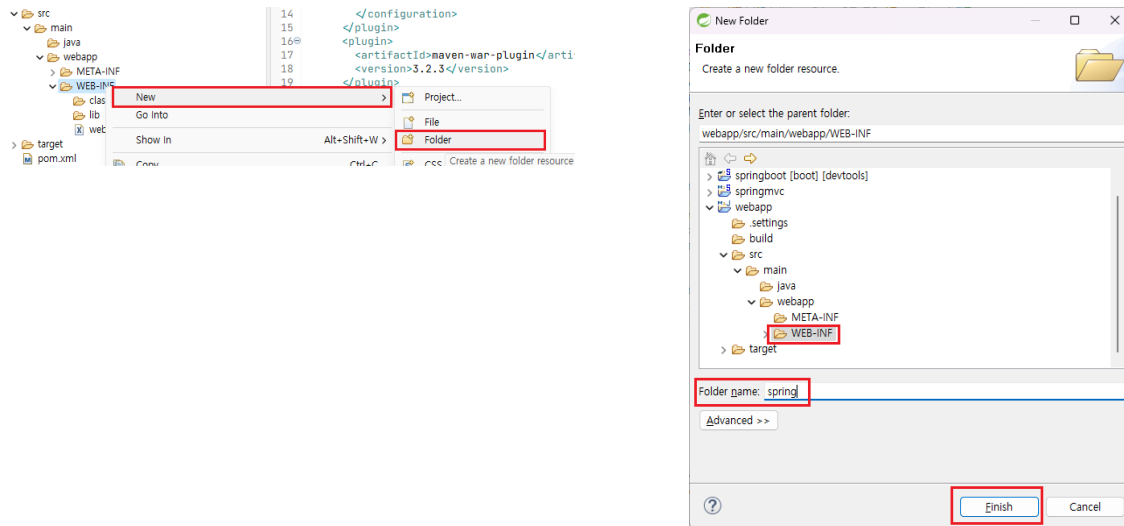
- pom.xml 수정
 - Maven 설정 후 저장한다.
 - Maven > Update Project



Spring

프로젝트 설정

- ◆ Spring Framework 환경 설정
 - src > main > webapp > WEB-INF > "spring" 폴더 생성한다.

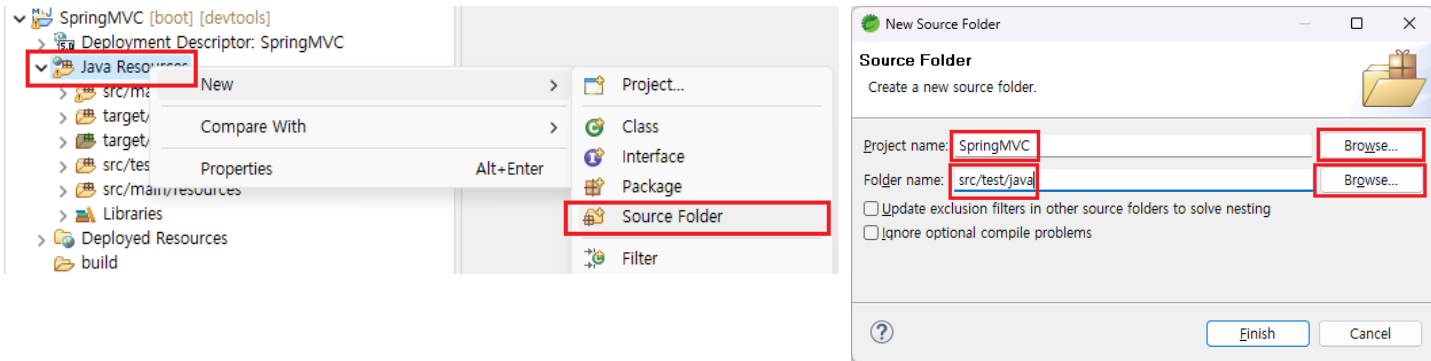


- /webapp/src/main/webapp/WEB-INF/spring 폴더에 spring framework 환경설정 파일을 생성한다.
- src > main > webapp > WEB-INF > "views" 폴더 생성한다.
- web.xml에 스프링 환경 설정을 등록한다.

Spring

프로젝트 설정

- "test" 폴더 추가
 - "%Project Root%/src/test/java", "%Project Root%/src/test/resources" 폴더 생성
 - Java Resources {M:R} > New > Source Folder
 - Project name: 프로젝트 선택
 - Folder name: src/test/java



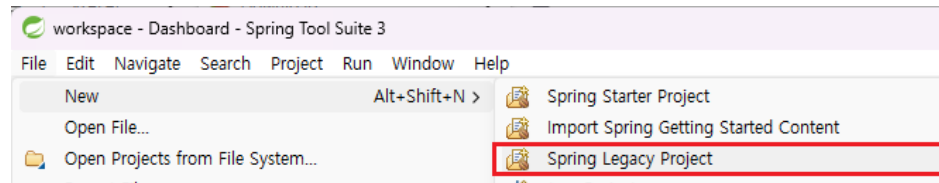
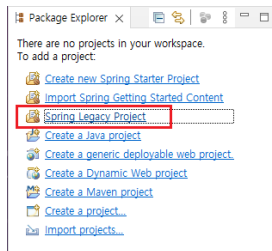
- "resources" 폴더 추가
 - "%Project Root%/src/main/resources" 폴더 생성
 - Java Resources {M:R} > New > Source Folder
 - Project name: 프로젝트 선택
 - Folder name: src/main/resources
- Project > Properties > Java Build Path > Source 에서 확인
- 이미 등록된 경우 폴더만 생성해 준다.

Spring

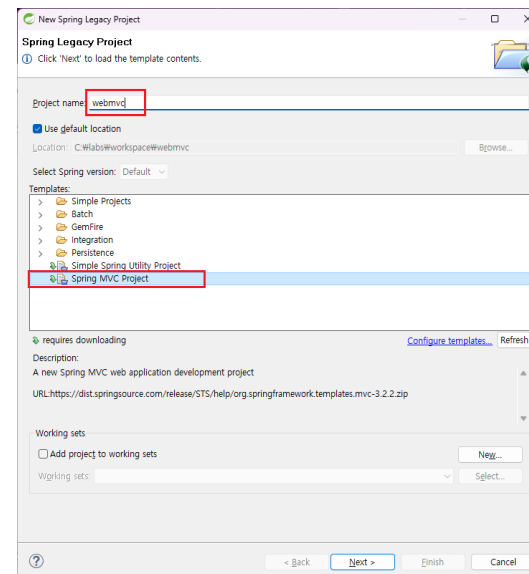
프로젝트 설정

◆ Spring Legacy Project

- ◆ Package Explorer > Spring Legacy Project
- ◆ File > New > Spring Legacy Project

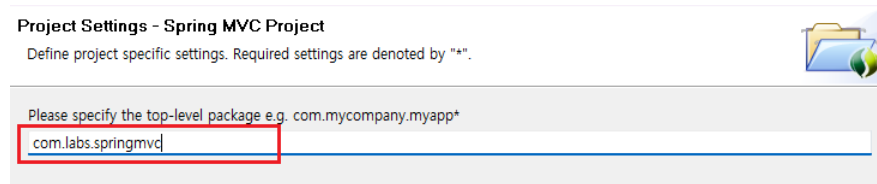


- ◆ Templates > Spring MVC Project 선택
 - Project Name : springmvc



I 프로젝트 설정

- ◆ top-level Package
 - com.labs.springmvc



- Project Explorer를 활성화한다. (선택)
 - Window > Show View > Project Explorer
- ◆ 공통 설정 확인을 진행한다.

Spring

프로젝트 설정

◆ Spring Boot

◆ Spring_INITIALIZER

- 프로젝트 구성 요소들을 선택하여 생성된 gradle, maven 설정 파일을 .zip 파일로 다운로드 받아 импорт(import)하여 프로젝트를 생성하는 방법이다.
- <https://start.spring.io/>
 - Group : 프로젝트의 유일한 식별자이어야 한다. 자바 패키지 명명법을 이용한다.
 - Artifact : 버전을 생략한 빌드된 결과물(파일)명으로 특수 문자 없는 소문자를 이용한다. 프로젝트명으로 이용된다.
 - Name : 출력될 프로젝트명을 지정한다. 일반적으로 Artifact와 동일하게 지정한다.
- Artifact : springboot

The screenshot displays the Spring Initializr web interface. It features a sidebar with a hamburger menu and a main content area. The main area is divided into several sections: Project, Language, Dependencies, Project Metadata, and a list of recommended dependencies. The Project section shows 'Project' as 'Gradle - Groovy' and 'Language' as 'Java'. The Dependencies section lists 'Spring Boot DevTools', 'Lombok', 'Spring Web', 'Thymeleaf', 'Spring Security', 'OAuth2 Client', and 'MyBatis Framework'. The Project Metadata section includes fields for Group, Artifact, Name, Description, Package name, and Packaging. The bottom of the page has a 'GENERATE' button and social media links.

Project

☒ Gradle - Groovy ☐ Gradle - Kotlin ☐ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Dependencies ADD DEPENDENCIES... CTRL + B

Spring Boot DevTools DEVELOPER TOOLS
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Lombok DEVELOPER TOOLS
Java annotation library which helps to reduce boilerplate code.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Thymeleaf TEMPLATE ENGINES
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

Spring Security SECURITY
Highly customizable authentication and access-control framework for Spring applications.

OAuth2 Client SECURITY
Spring Boot integration for Spring Security's OAuth2/OpenID Connect client features.

MyBatis Framework SQL
Persistence framework with support for custom SQL, stored procedures and

Project Metadata

Group:

Artifact:

Name:

Description:

Package name:

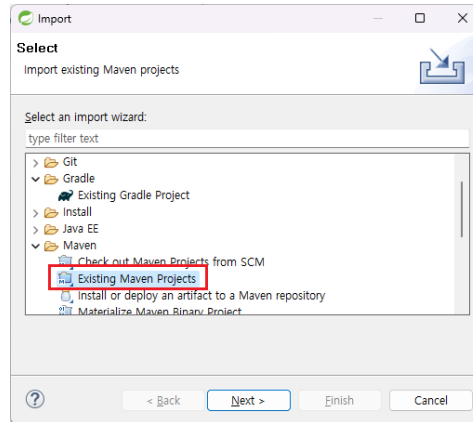
Packaging: ☒ Jar ☐ War

Java: ☐ 19 ☐ 17 ☒ 11 ☐ 8

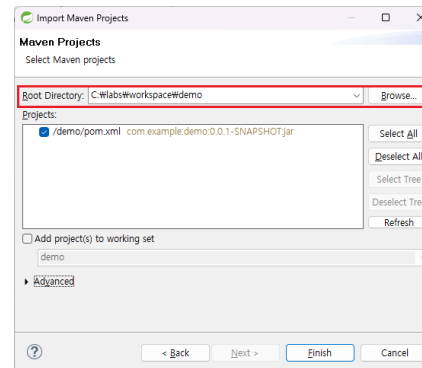
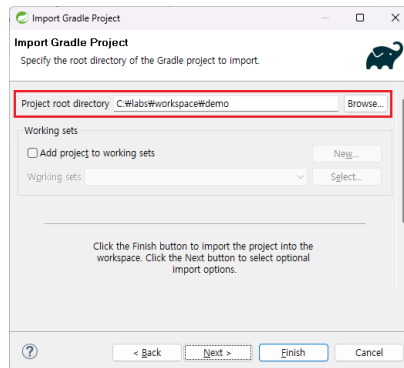
GENERATE CTRL + G **EXPLORE** CTRL + SPACE **SHARE...**

I 프로젝트 설정

- 다운로드된 zip 파일을 워크스페이스(workspace) 폴더에 풀어준다.
- Package Explorer > Import Projects..., File > Import... 를 실행한다.



- Gradle, Maven 중 설정에 적절한 방법을 선택한다.

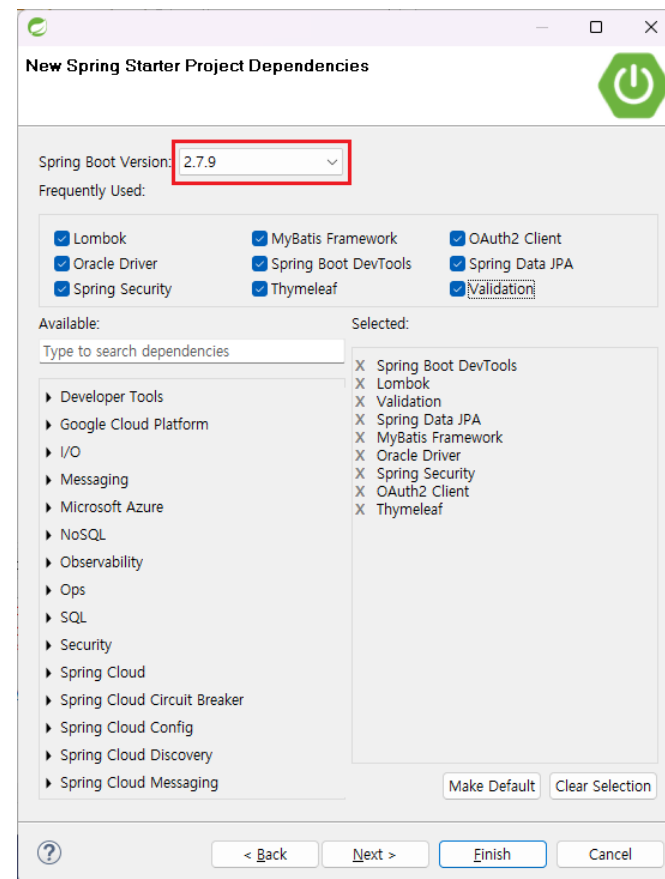
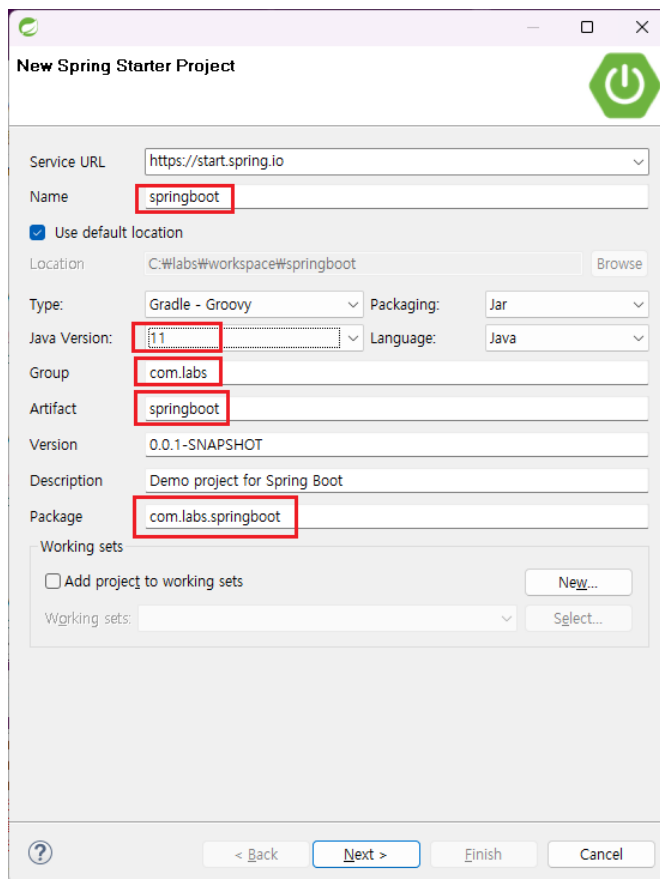


- Project Explorer를 활성화한다. (선택)
 - Window > Show View > Project Explorer

Spring

I 프로젝트 설정

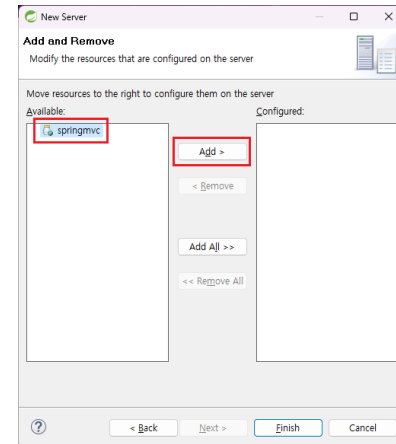
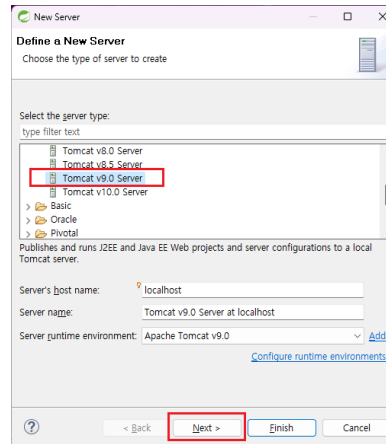
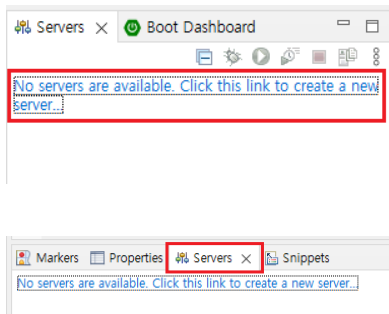
- ◆ Spring Starter
 - STS 에서 이용 가능하다.
 - Package Explorer > Create new Spring Starter Project
 - File > New > Spring Starter Project



프로젝트 설정

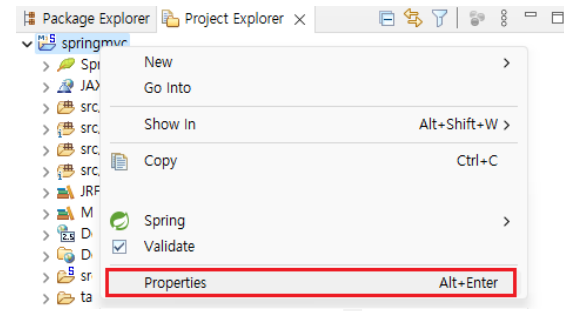
◆ 공통 설정 확인

- ◆ 각 프로젝트 유형별로 진행한다.
- ◆ 실행 서버등록
 - Dynamic Web Project, Spring Legacy Project 에서 진행한다. Spring Starter는 선택



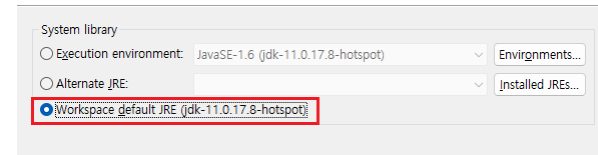
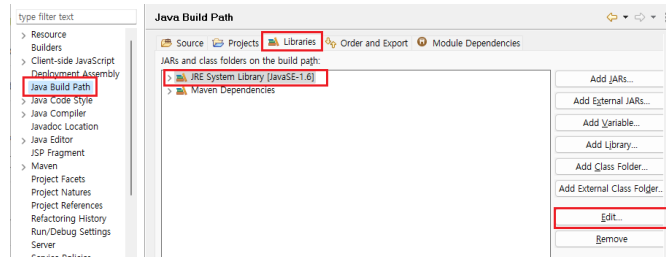
◆ 설정 확인

- Project > Properties, Project {M:R} > Properties

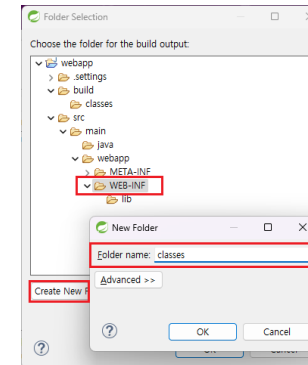
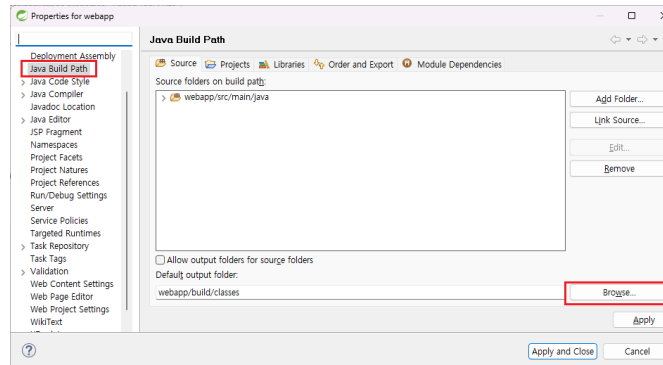


프로젝트 설정

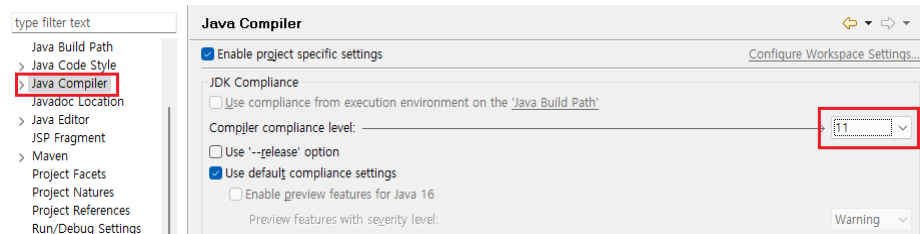
- Java Build Path > Libraries



- Java Build Path > Source
 - Dynamic Web Project 에서 진행

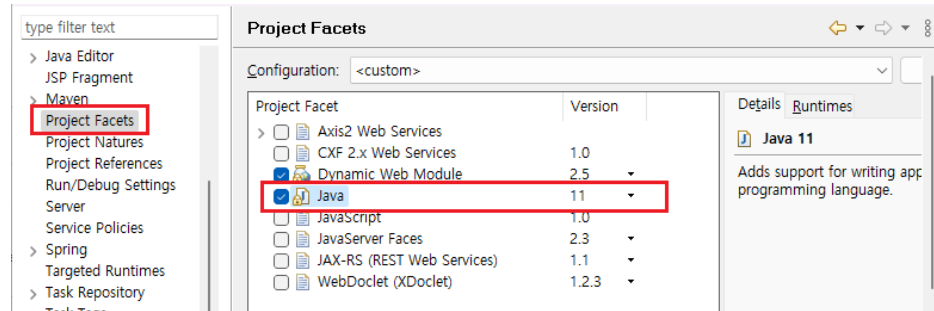


- Java Compiler

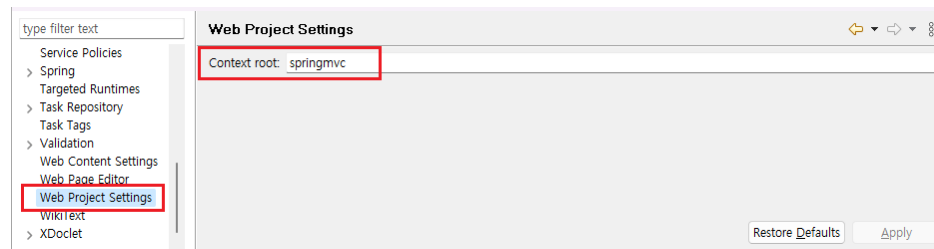


프로젝트 설정

- Project Facets

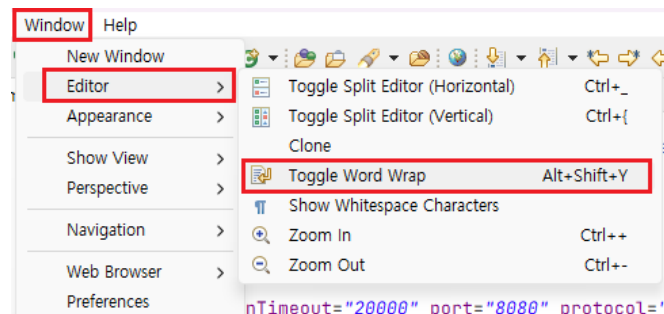


- Web Project Settings



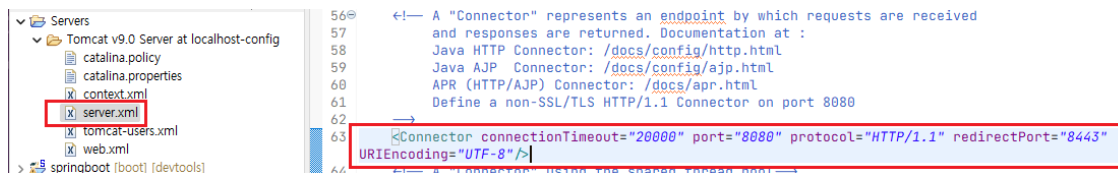
◆ Word Wrap

- Code Editor 선택 후 설정한다.



I 프로젝트 설정

- ◆ Tomcat 서버 및 Web.xml에서 한글 처리 설정
 - Dynamic Web Project, Spring Legacy Project 에서 진행한다
 - GET
 - server.xml, %tomcat%/conf/server.xml
 - <Connector protocol="HTTP/1.1" ... URIEncoding="UTF-8" /> 추가



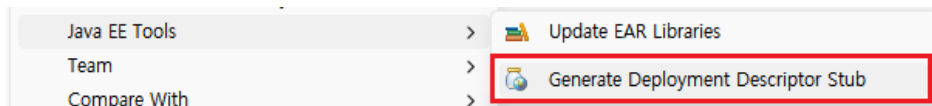
- POST
 - src/main/webapp/WEB-INF/web.xml

```
<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>
```

I 프로젝트 설정

```
<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
```

- ◆ web.xml이 없는 경우
 - Java EE Tools 메뉴를 이용하여 생성한다.

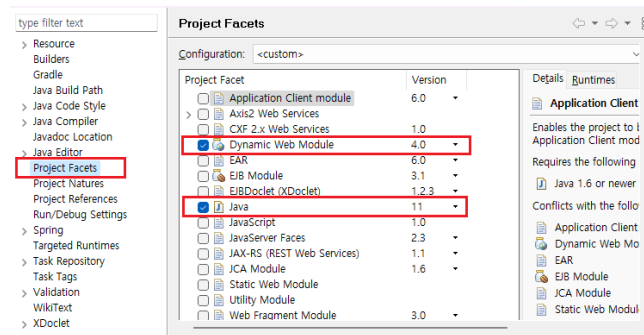


Spring

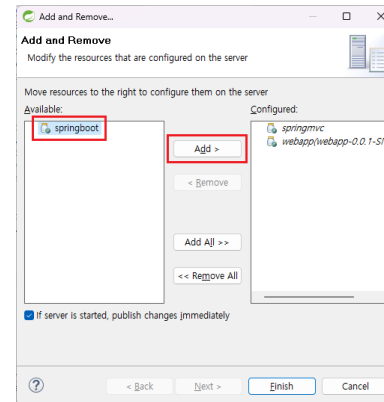
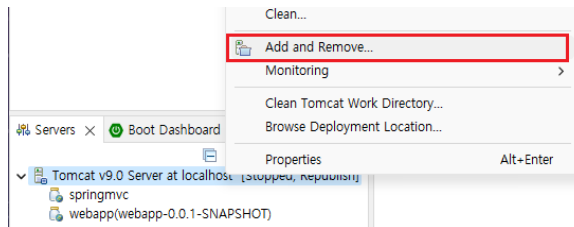
프로젝트 설정

◆ 기타

- ◆ SpringBoot 외부 Tomcat 연동
 - Project > Properties > Project Facets 설정



- context 등록



Spring

■ 참고자료

- ◆ <https://guide.ncloud-docs.com/docs/sso-integration-info>
- ◆ <https://docs.spring.io/spring-framework/docs/current/reference/html/core.html>
- ◆ <https://spring.io/projects/spring-framework>
- ◆ <https://spring.io/projects/spring-boot>
- ◆ https://ko.wikipedia.org/wiki/스프링_프레임워크
- ◆ <https://www.geeksforgeeks.org/pojo-vs-java-beans/>
- ◆ https://ko.wikipedia.org/wiki/의존성_주입
- ◆ https://ko.wikipedia.org/wiki/관점_지향_프로그래밍
- ◆ https://namu.wiki/w/객체_지향_프로그래밍/원칙

Spring