

1. 개발 환경 구축

관리의 편의를 위해 하나의 폴더를 정해서 다운 설치하는 것이 좋다. 프레임워크 사용 시 폴더 경로에 한글이 있을 경우 문제가 될 수 있는 경우가 있으므로 폴더 명에 한글이 포함되지 않도록 한다.

1) JDK 다운 설치 → Java 수업 시 설치했을 것

2) Tomcat 다운 설치 → 서버 수업 시 설치했을 것

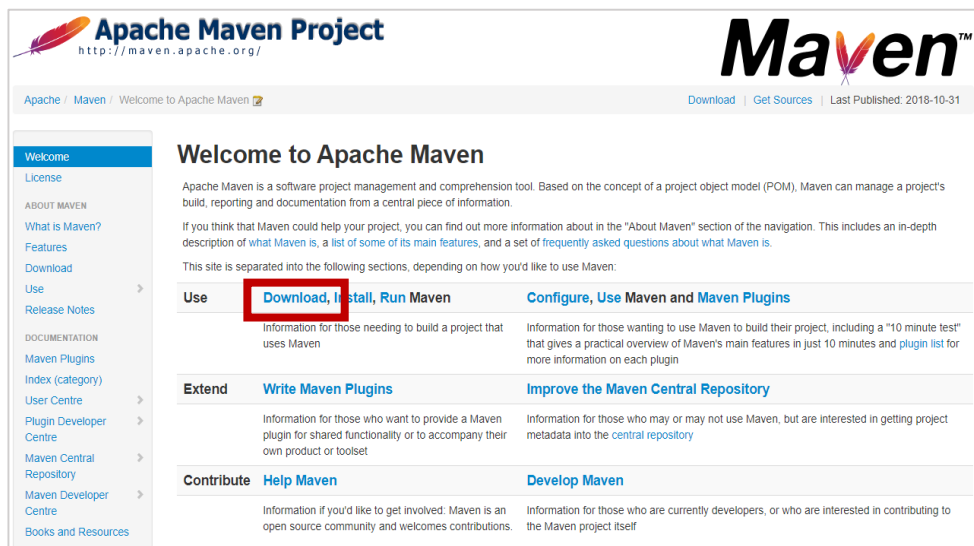
3) Maven 다운 설치

Maven이란 자바용 프로젝트 관리 도구로, **Project Object Model(POM) XML** 문서를 통해 해당 프로젝트의 버전 정보 및 라이브러리 정보들을 통합하여 관리하는 프레임워크이다. 일반적으로 프로젝트는 개발자가 필요한 라이브러리를 직접 찾아서 다운로드 및 추가해야 하지만 Maven을 사용하면 pom.xml 문서에 내가 사용하고자 하는 라이브러리를 등록하여 자동으로 프로젝트에 추가되게 하여 라이브러리 관리 및 버전 관리의 편리성을 제공해준다.

라이브러리를 다운 설치 받을 폴더를 명시적으로 지정하여 관리하는 것이 좋다.

→ Apache Maven 사이트에 접속하여 다운로드

<https://maven.apache.org/> 로 접속하여 다운로드



Downloading Apache Maven 3.8.6

Apache Maven 3.8.6 is the latest release and recommended version for all users.

The currently selected download mirror is <https://d1cdn.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are [backups](#). You may also consult the [complete list of mirrors](#).

Other mirrors:

Files

↓ Scroll Down..

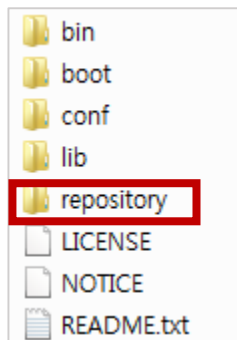
Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [installation instructions](#). Use a source archive in order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles against the public [KEYS](#) used by the Apache Software Foundation.

	Link	Checksums
Binary tar.gz archive	apache-maven-3.8.6-bin.tar.gz	apache-maven-3.8.6-bin.tar.gz.sha512
Binary zip archive	apache-maven-3.8.6-bin.zip	apache-maven-3.8.6-bin.zip.sha512
Source tar.gz archive	apache-maven-3.8.6-src.tar.gz	apache-maven-3.8.6-src.tar.gz.sha512
Source zip archive	apache-maven-3.8.6-src.zip	apache-maven-3.8.6-src.zip.sha512

* 다운로드 받은 압축 파일을 C:\wdev 폴더에 압축 해제

→ 라이브러리들이 다운로드 되어 저장될 폴더 생성

압축해제 한 폴더에 'repository'라는 저장소 역할의 새 폴더 생성



→ settings.xml 파일 수정

conf 폴더에 접근하여 settings.xml 파일을 열어 주석 처리된 <localRepository>란 부분을 찾아 복사하여 밖으로 뺀 뒤, 저장소로 이용하고자 하는 폴더(위에서 만든 repository경로)로 설정한다.

```
<!-- localRepository
 | The path to the local repository maven will use to store artifacts.
 |
 | Default: ${user.home}/.m2/repository
<localRepository>/path/to/local/repo</localRepository>
-->
<localRepository>D:\wdev\workspace\apache-maven-3.5.4\repository</localRepository>
```

4) STS(Spring Tool Suite) 설치하기

STS란 Spring Tool Suite의 약자로, **Spring Framework를 사용하기 위한 개발 툴**을 말한다. 일반적으로는 별도의 설치 도구를 통해 설치하여 사용 가능하고, 이클립스 IDE에서 제공하는 STS 플러그 인을 통해 간단히 설치할 수 있다.

* 하단에 두 개의 방법 중 1개로!! (방법1 권장)

➔ **방법1: STS3 직접 설치하기 (기존에 spring 공식사이트에서 제공 되었지만 바뀜...)**

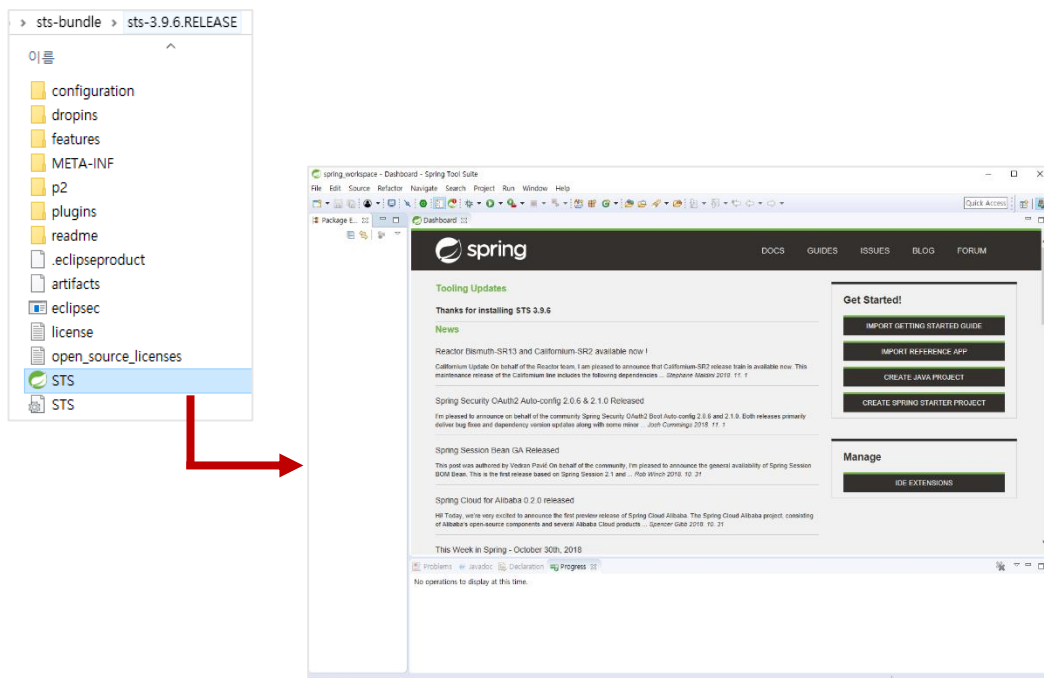
STS3 설치 사이트 : <https://github.com/spring-projects/toolsuite-distribution/wiki/Spring-Tool-Suite-3>

Spring Tool Suite 3.9.15 (New and Noteworthy)

full distribution on Eclipse 4.18

- https://download.springsource.com/release/STS/3.9.15.RELEASE/dist/e4.18/spring-tool-suite-3.9.15.RELEASE-e4.18.0-win32-x86_64.zip
- https://download.springsource.com/release/STS/3.9.15.RELEASE/dist/e4.18/spring-tool-suite-3.9.15.RELEASE-e4.18.0-macosx-cocoa-x86_64.dmg
- https://download.springsource.com/release/STS/3.9.15.RELEASE/dist/e4.18/spring-tool-suite-3.9.15.RELEASE-e4.18.0-linux-gtk-x86_64.tar.gz

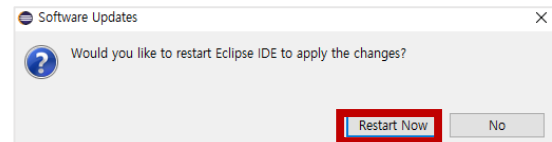
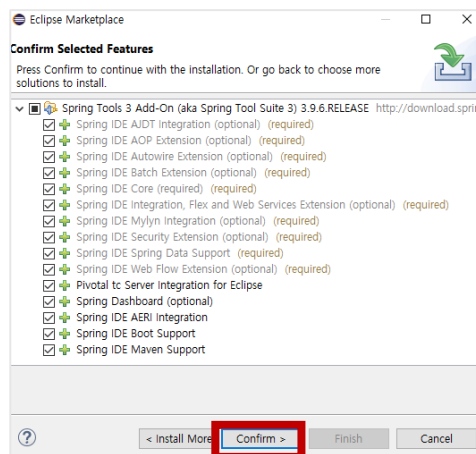
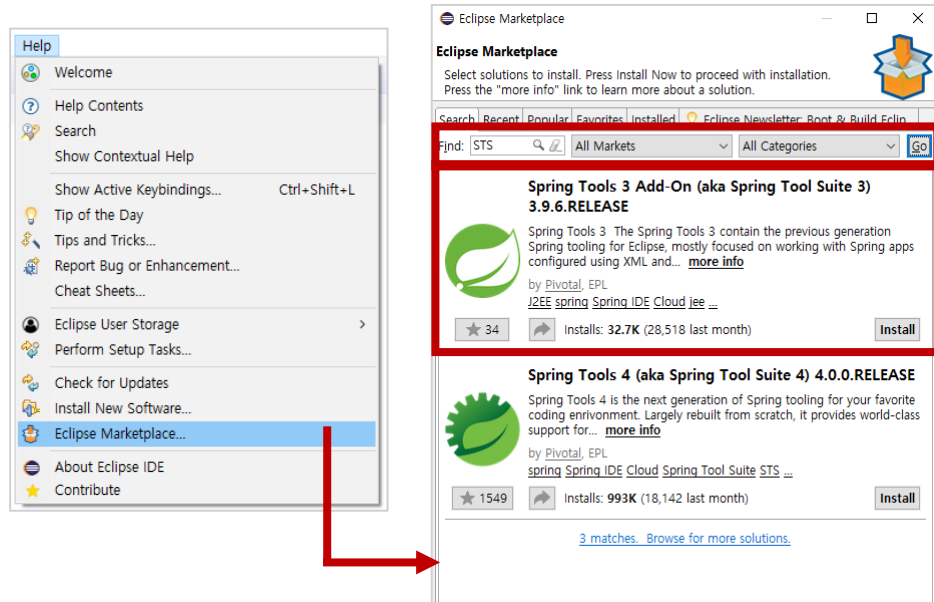
* 다운로드 된 zip 파일 C:\w에 압축 해제 (압축해제 전 zip파일명 짧게 줄여야 됨)



* STS 프로그램을 실행했을 때 위와 같은 화면이 보인다면 정상적으로 설치됨

➔ 방법2: Eclipse STS 플러그인을 통해 설치하기

기존의 이클립스에서 [Help] – [Eclipse Marketplace...] 클릭하고 검색 창에 'STS' 입력 하여 검색 후 설치



* STS 설치 완료 후 재실행한다.

➔ 보통 방법2를 통해 간단하게 설치 하지만 플러그인이 제대로 설치가 안되거나 호환이 맞지 않는 경우 방법1과 같은 방식으로 진행한다.

(또한 이클립스 버전에 따라 Market Place에 검색이 안됨!!)

5) 데이터베이스 구축

➔ spring/spring 계정 생성 후 수업시간에 배포된 Table Script.sql 실행

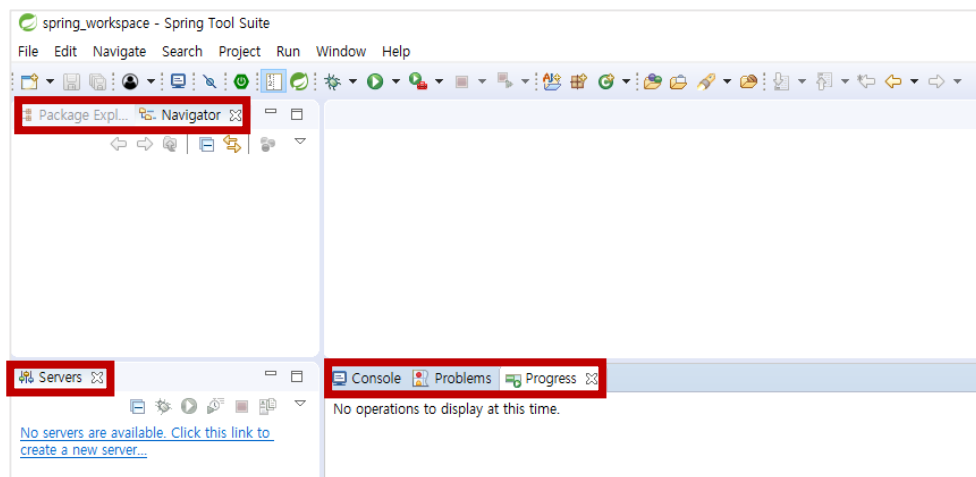
2. Workspace 환경 설정 및 프로젝트 환경 설정

1) Workspace 환경 설정

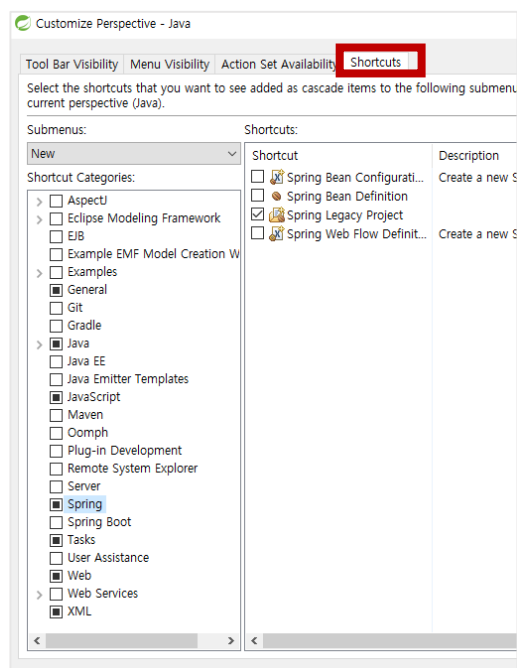
→ Spring 프로젝트 작업 시 필요한 환경 셋팅

- [Window] – [Show View] 를 통해 아래와 같이 적용

'Package/Project/Spring Explorer', 'Navigator', 'Servers', 'Console', 'Problems', 'Progress'



- [Window] – [Perspective] – [Customize Perspective] – [Shortcuts] 탭을 통해 필요한 메뉴들 선택 (New 했을 때 보여질 목록들)



→ 인코딩 설정

[Window] – [Preferences]에 접근하여 아래 부분들 모두 'UTF-8'로 적용

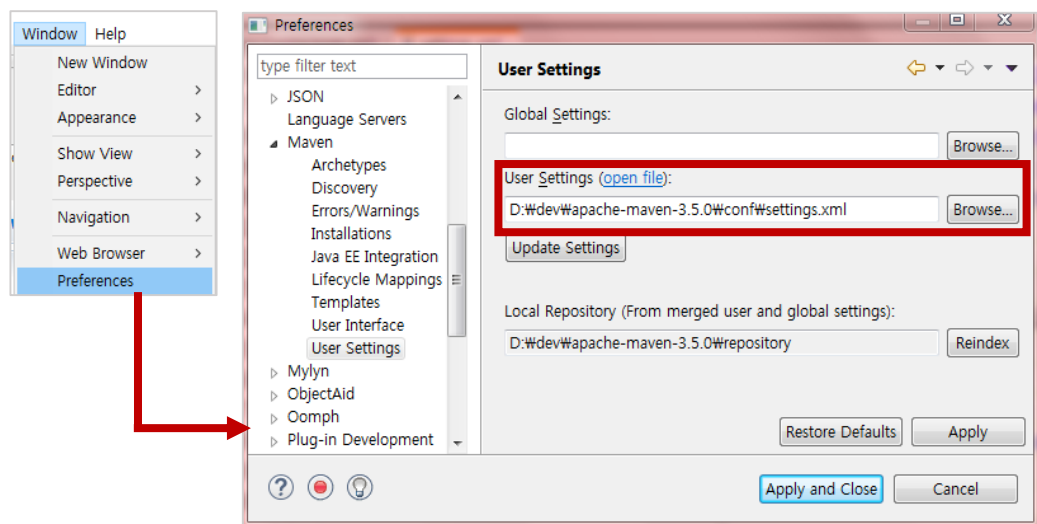
- [General] – [Workspace] – [Workspace]
- [General] – [Editors] – [Text Editors] – [Spelling]
- [JSON] – [JSON Files]
- [Web] – [CSS Files] / [HTML Files] / [JSP Files]
- [XML] – [XML Files]

→ Java > Code Templates 수정

→ JSP > Templates 수정

→ Eclipse(STS)와 Maven 연동

[Window] – [Preferences] → [Maven] – [User Settings]에 접근하고 [User Settings]의 값을 <localRepository> 경로를 지정한 conf 폴더의 settings.xml의 경로로 변경한다.



* 다음과 같이 세팅해 놓으면 앞으로 라이브러리들이 내가 만든 해당 그 repository 폴더에 자동으로 다운로드됨.

➔ **Server Runtime Environments 설정**

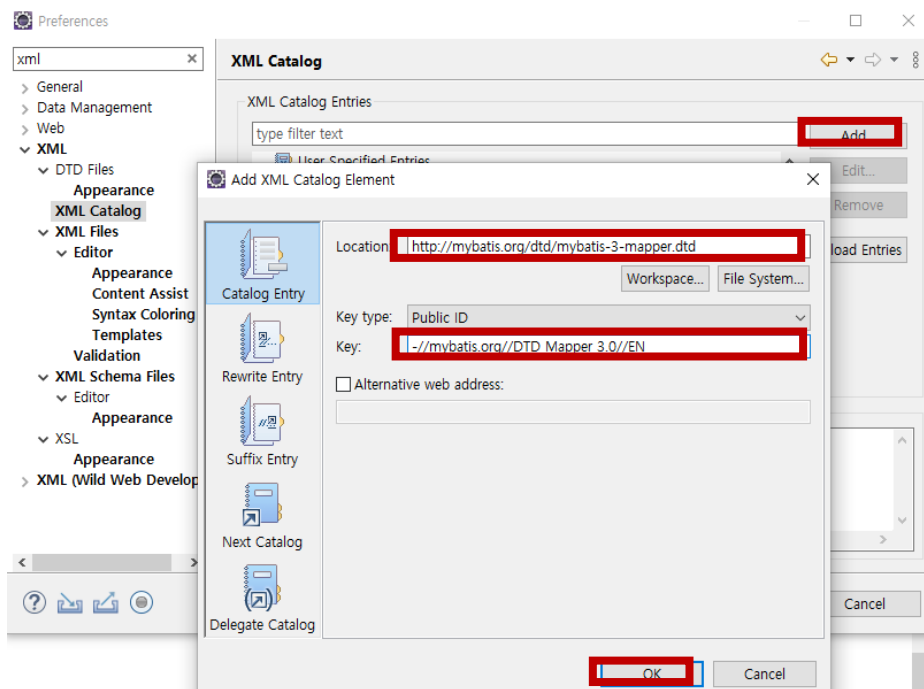
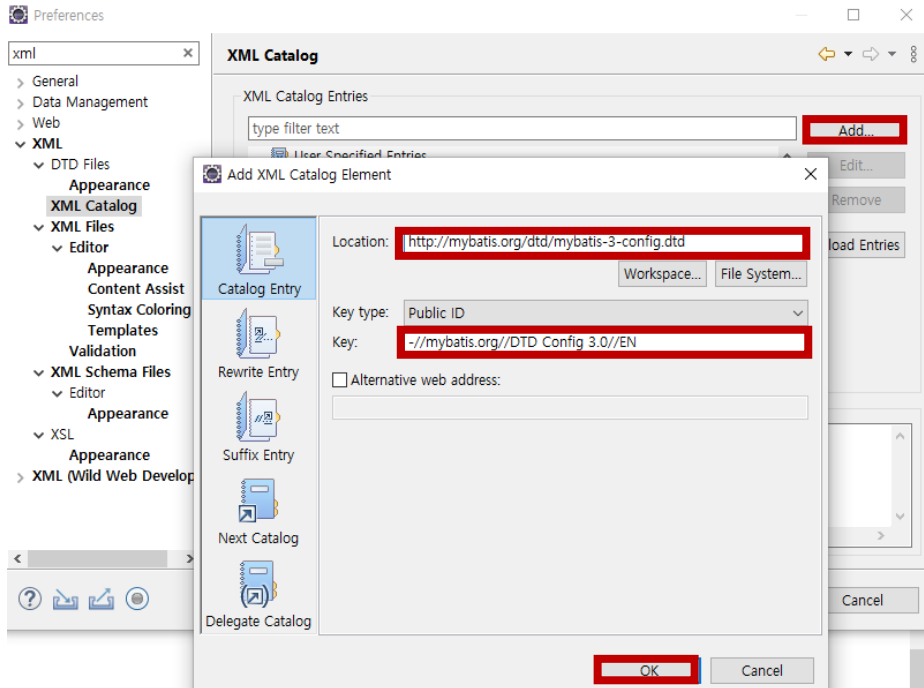
[Window] – [Preferences] → [Server] – [Runtime Environments] 을 통해 톰캣 환경을 구축해준다.

➔ **Local Server 구축(생성)**

- Port 변경
- Server modules without publishing 체크!!

→ XML Catalog 등록

마이바티스 관련 xml 파일을 만들기 위해 필요한 dtd 미리 설정해두기



2) 프로젝트 생성

→ Spring Legacy Project 생성

[New] – [Spring Legacy Project] → 프로젝트 이름 입력

Templates : 'Spring MVC Project' 템플릿 선택

[Next] > 최상위 패키지 지정 : **최소 3개 이상의 레벨**로 지정할 것

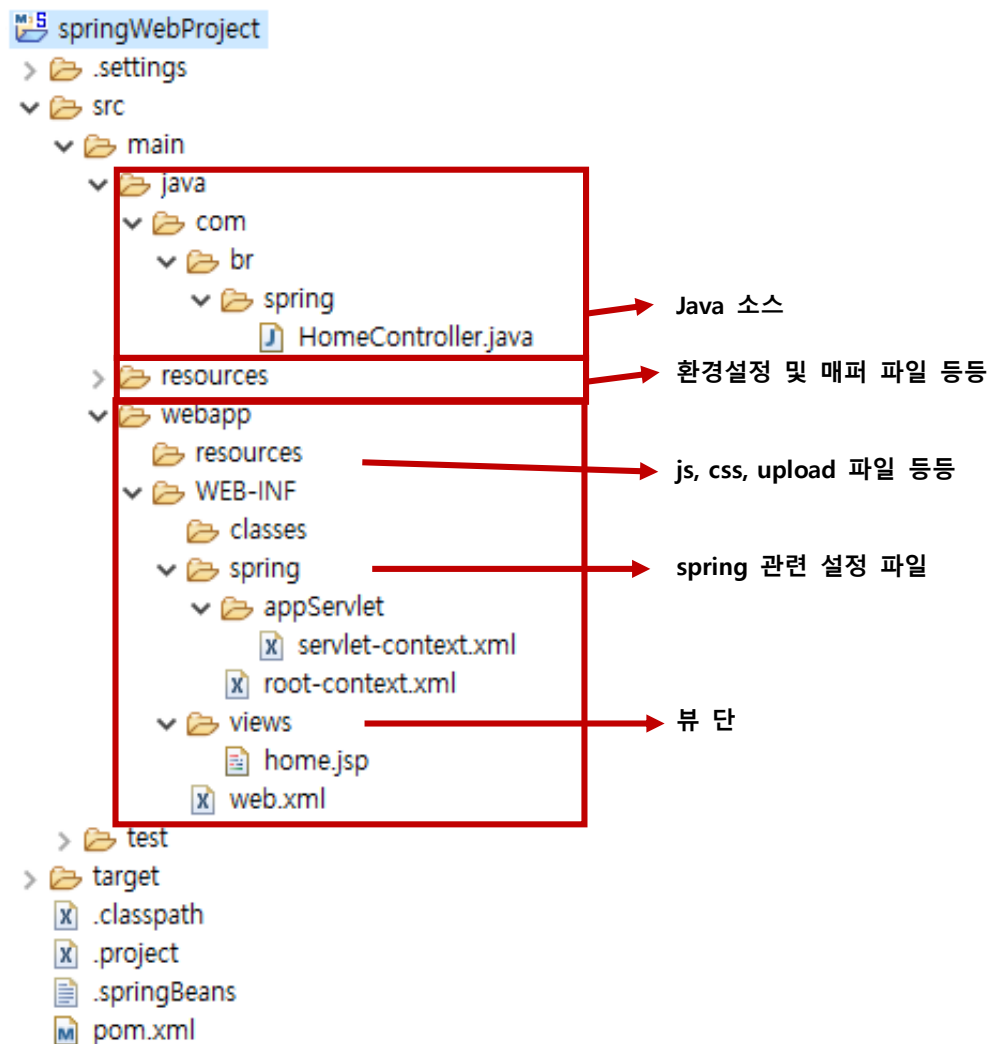
3번째 레벨이 ContextPath가 된다.

* 프로젝트 생성시 기본적으로 새로 생성된 프로젝트에 필요한 라이브러리를 인터넷에서 다운받게 되는데 꽤 시간이 소요된다. (Maven이 자동으로 해줌)

→ progress 바를 통해 확인 후 끝날 때 까지 대기

(제어권이 우리에게 있는게 아니라 프레임워크한테 있다.)

→ 아까 만든 repository 경로에 라이브러리들이 잘 다운로드 됐는지 확인!



→ 프로젝트 설정 변경

STS를 이용하여 'Spring MVC Project' 프로젝트를 생성하면 JRE 버전도 맞지 않고 서버 라이브러리도 등록되어 있지 않다. → 설정을 수정해 줘야 된다.

프로젝트에서 오른쪽버튼 - [Properties]

- [Project Facets] 탭

> Java 버전 1.8 또는 설치된 버전으로 수정

> Runtimes 탭 - 등록된 서버로 체크 > Apply

- [Java Build Path] 탭 - [Libraries] 탭

> Tomcat, JRE, Maven 등록 확인 함 > Apply and Close

→ pom.xml 파일에서 자바 버전 변경 및 Spring 버전을 가장 최신 버전으로 변경

스프링 버전 확인 : <http://projects.spring.io/spring-framework>

```
<properties>
<java-version>11</java-version>
<org.springframework-version>5.3.22</org.springframework-version>
<org.aspectj-version>1.6.10</org.aspectj-version>
<org.slf4j-version>1.6.6</org.slf4j-version>
</properties>
```

pom.xml을 수정하고 나서 어느 정도 시간이 지나면, Package Explorer 뷰에서 Java Resources/Libraries/Maven Dependencies에 스프링관련 라이브러리들 버전이 일괄적으로 변경된 것을 확인할 수 있다.

→ 서버에 프로젝트 add 하고 실행 확인

- webapp 폴더에 index.jsp 페이지 생성
- jstl 관련 에러 발생 시 jstl 관련 라이브러리(4개)를 수동으로 WEB-INF/lib 폴더에 추가할 것

3) 추가적으로 필요한 라이브러리 등록 (pom.xml)

기본적으로 필요한 라이브러리들은 프로젝트 생성시 Maven이 자동으로 다운받아 준다. 하지만 추가적으로 다뤄야 할 라이브러리들은 pom.xml을 통해 우리가 등록을 해줘야 된다.

* 라이브러리 검색 사이트 (북마크 추가할 것!!) : <https://mvnrepository.com/>

해당 사이트 접속 불가시 : vpn 이용해서 접속할 것

➔ Oracle 드라이버와 MyBatis 라이브러리 내려 받기 (DB 관련한 라이브러리들)

<dependencies> 위에 <repositories>를 이용해 사설저장소 등록

```
<repositories>
  <repository>
    <id>Datanucleus</id>
    <url>http://www.datanucleus.org/downloads/maven2/</url>
  </repository>
</repositories>
```

<dependency> 이용해서 오라클, 마이바티스 관련 라이브러리들 추가

```
<!-- 1_1. ojdbc6 라이브러리 -->
<dependency>
  <groupId>oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.3</version>
</dependency>
<!-- 1_2. 스프링에서 JDBC 기능을 사용하기 위한 라이브러리 -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
<!-- 1_3. 순수 myBatis 라이브러리 -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.9</version>
</dependency>
<!-- 1_4. 스프링에서 마이바티스를 사용하기 위한 라이브러리 -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.3.2</version>
</dependency>
<!-- 1_5. 커넥션 풀을 사용하기 위한 라이브러리 -->
<dependency>
  <groupId>commons-dbcp</groupId>
  <artifactId>commons-dbcp</artifactId>
  <version>1.4</version>
</dependency>
```