

Spring Framework

3강

JSP에 스프링조립하기

JSP에 Maven을 이용한 스프링 조립하기

스프링을 처음 다루는 초보자가 이해하기 힘든 부분일 수 있습니다.

JSP에 Maven을 붙이고, Maven을 통해 **스프링 프레임워크를 추가**하는 방법입니다.

스프링 프로젝트부터는 어떻게 생성하냐 에 따라 **프로젝트 구조가 다를 수 있습니다.**

프로젝트 초기 구축 환경에 따라 많이 달라질 수 있습니다.

수동으로 간단하게 구축해보고, 자동 생성을 진행하도록 하겠습니다.

JSP에 빌드툴을 사용하는 방법은 대표적으로 2가지입니다.

1. Maven을 사용하는 방법
2. Gradle을 사용하는 방법

스프링 설정파일을 사용하는 방법은 대표적으로 2가지입니다.

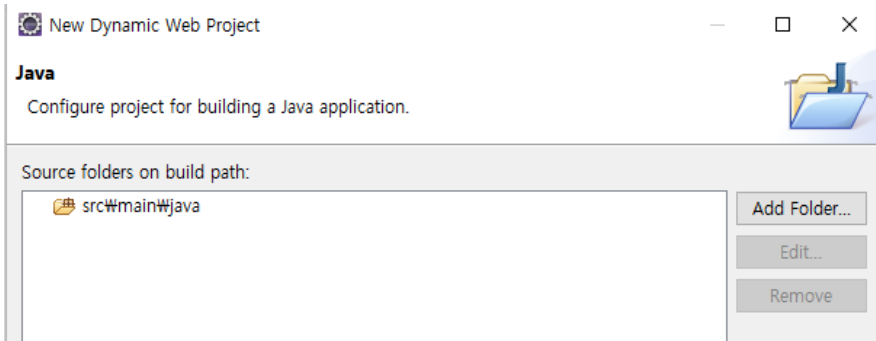
1. xml을 이용하는 방법
2. 자바코드를 이용하는 방법

JSP에 Maven을 이용한 스프링 조립하기

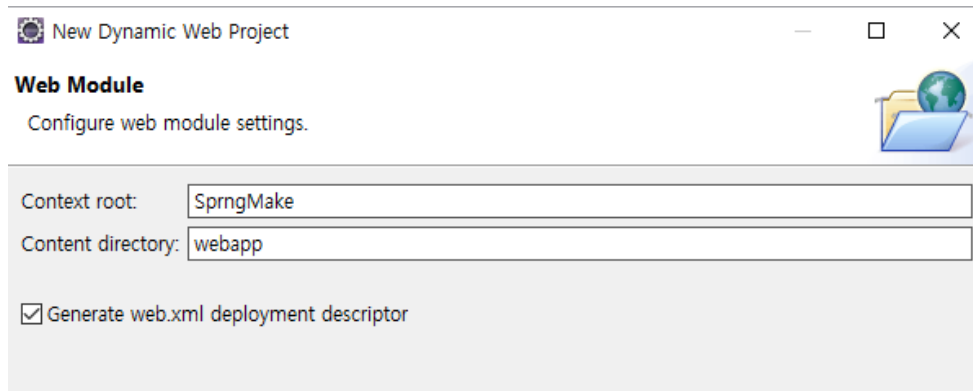
1)

Dynamic Web Project를 생성합니다.

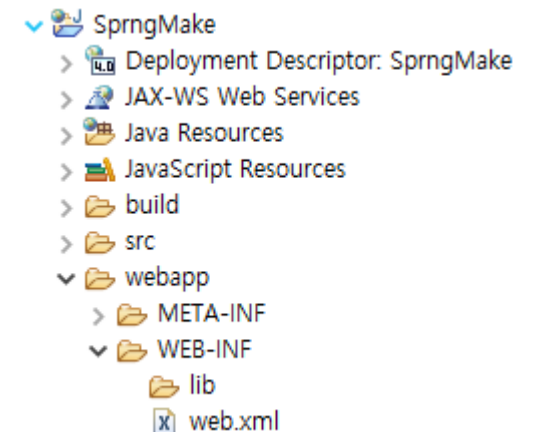
next버튼을 누르고 기본 파일 폴더명을 src/main/java로 변경합니다.



Content director를 webapp로 변경합니다.
아래 web.xml생성 버튼도 체크합니다.



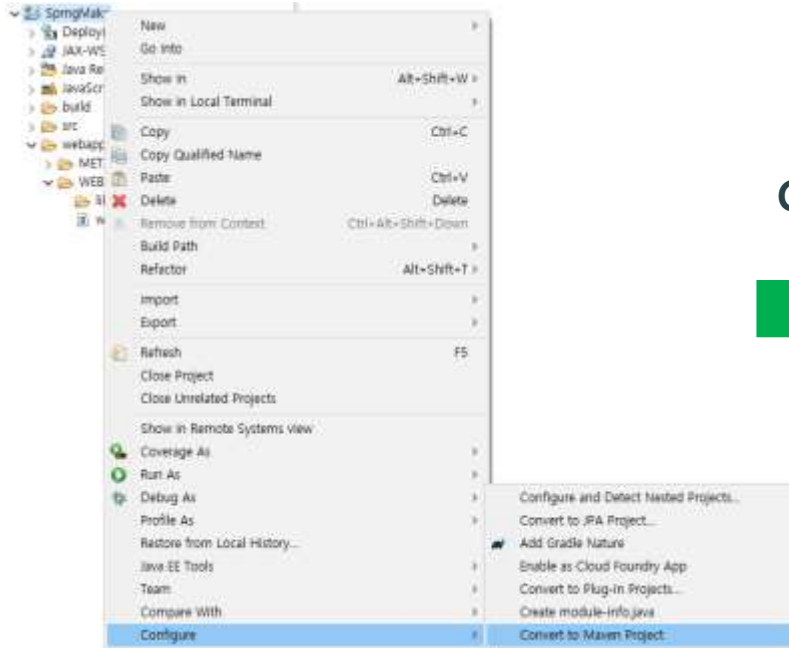
완성된 웹프로젝트



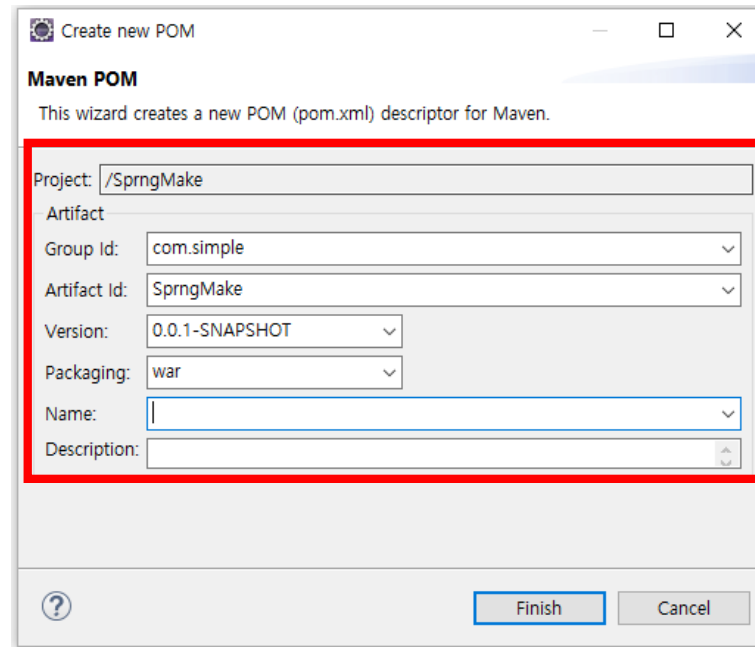
JSP에 Maven을 이용한 스프링 조립하기

2)

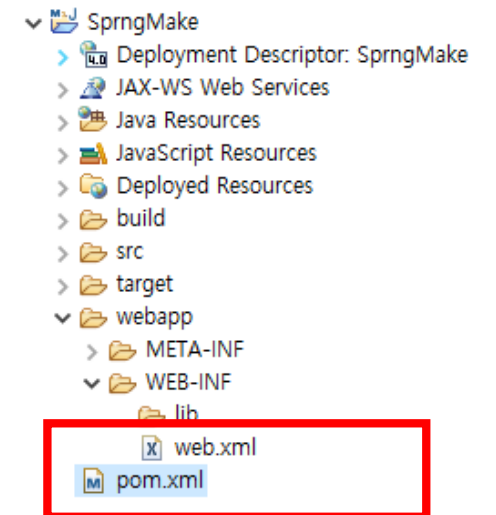
웹프로젝트 우 클릭 Covert to Maven Project를 클릭합니다



Group Id는 프로젝트를 생성하는 회사 등의 고유 이름입니다



모양이 변경되면서 Maven을 다룰 수 있는 pom.xml이 추가됩니다



JSP에 Maven을 이용한 스프링 조립하기

3) 완성된 pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.simple</groupId>
    <artifactId>SprngMake</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>war</packaging>

    <!-- 메이븐 빌드 툴 -->
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
            <plugin>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.2.3</version>
                <configuration>
                    <warSourceDirectory>webapp</warSourceDirectory>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

<https://mvnrepository.com/>

다음 필요한 라이브러리를 순서대로 추가합니다

Maven이 라이브러리를 제대로 받아오지 못할 경우

그럴 때는 아래의 경로에 가서 repository 폴더를 지우고 다시 받으면 됩니다.

₩원도우사용자계정₩.m2₩repository

JSP에 Maven을 이용한 스프링 조립하기

다음 필요한 스프링 코드를 순서대로 추가합니다

1. 자바 버전 스프링 버전 변수 선언
2. dependencies태그 선언
3. 스프링코어 다운
4. 스프링webMVC 다운
5. web.xml에 스프링servlet 설정(프로젝트 최초 가동시 동작)
6. servlet.xml (서블릿 설정)

JSP에 Maven을 이용한 스프링 조립하기

web.xml 설정에 추가

```
<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/config/spring-servlet.xml
      /WEB-INF/config/DB-servlet.xml
    </param-value>
  </init-param>
</servlet>

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

JSP에 Maven을 이용한 스프링 조립하기

web.xml 설정에 추가

```
<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/config/spring-servlet.xml
      /WEB-INF/config/DB-servlet.xml
    </param-value>
  </init-param>
</servlet>

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


JSP에 Maven을 이용한 스프링 조립하기

spring-servlet 설정에 추가

<!-- 어노테이션을 활성화 하겠다는 의미 -->

```
<mvc:annotation-driven/>
```

<!-- 스프링만의 어노테이션을 활성화 한다는 의미(단, package속성에 기술된 경로의 파일을 읽어서 빈으로 생성한다 -->

```
<context:component-scan base-package="com.simple.controller"/>
```

<!-- 뷰리졸버 설정 -->

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
```

```
    <property name="prefix" value="/WEB-INF/views/" />
```

```
    <property name="suffix" value=".jsp" />
```

```
</bean>
```

<!-- 정적자원 맵핑설정 -->

```
<mvc:resources mapping="/resource/**" location="/resource/" />
```

JSP에 Maven을 이용한 스프링 조립하기

spring-servlet 핸들러 구축

```
<!-- 예시 - 프론트 컨트롤 핸들러 -->
<!-- 컨트롤러를 종류의 따라 다양하게 만들어 줄 수 있다. -->
<bean id="aaa" class="com.simple.controller.MainController" />

<!-- 컨트롤러를 맵핑시키는 핸들러 클래스 커스터마이징 -->
<bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="order" value="1" />
    <property name="mappings">
        <props>
            <prop key="/test/aaa">aaa</prop> <!-- 위에 만들어진 aaa객체 주입 ->
            <prop key="/test/bbb">aaa</prop>
        </props>
    </property>
</bean>
```