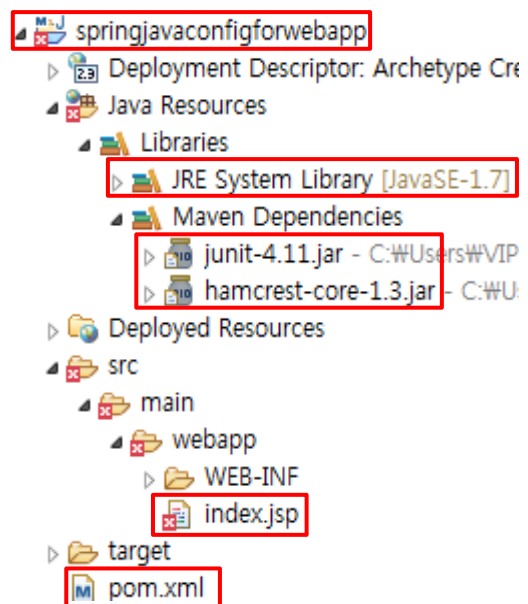
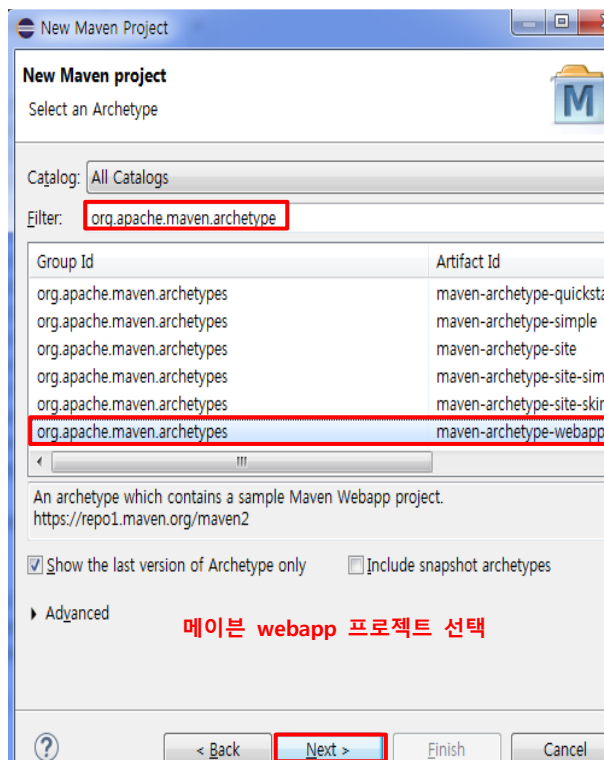


1. 유스케이스

- XML 설정(web.xml)대신 자바 기반 설정으로 개발
- **WebApplicationInitializer**을 상속해서 스프링 설정 만들기
- **web.xml**을 WebApplicationInitializer를 상속한 클래스로 대체
- **WebMvcConfigurerAdapter**를 상속한 SpringConfig 만들기
- SpringConfig: Internal **view resolver** 설정
- @EnableWebMvc: <mvc:annotation-driven /> -> dispatcher-servlet.xml
- @ComponentScan: <context:component-scan />
- @Configuration: Spring Configuration

web.xml	WebApplicationInitializer
<mvc:annotation-driven/>	@EnableWebMvc
<context:component-scan/>	@ComponentScan
Spring Configuration	@Configuration
SpringConfig extends WebMvcConfigurerAdapter	

2. 메이븐 프로젝트 만들기: springmvc 프로젝트 활용



springjavaconfigforwebapp/pom.xml

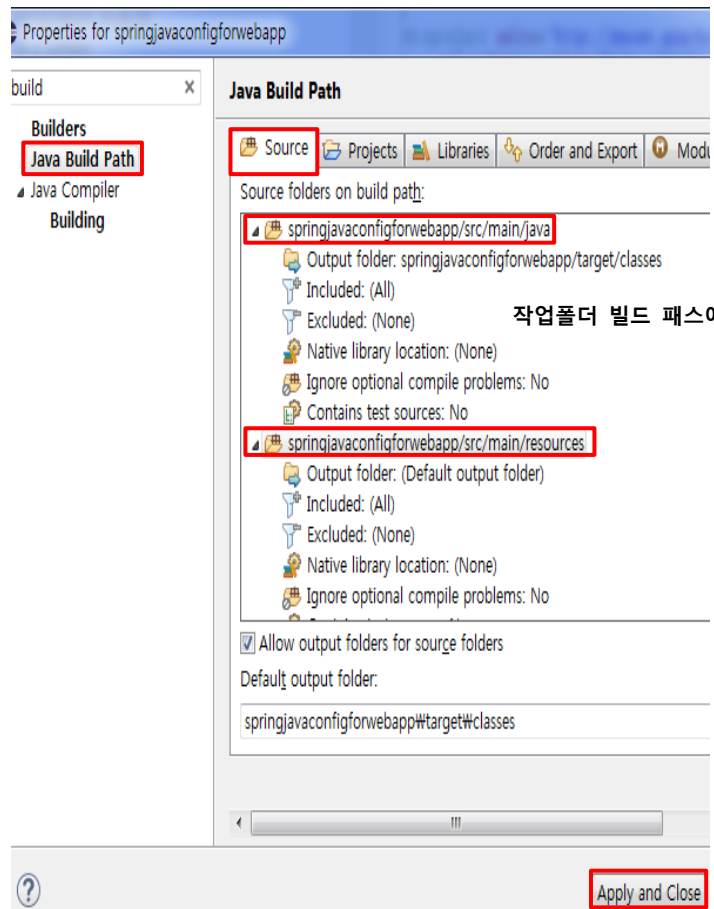
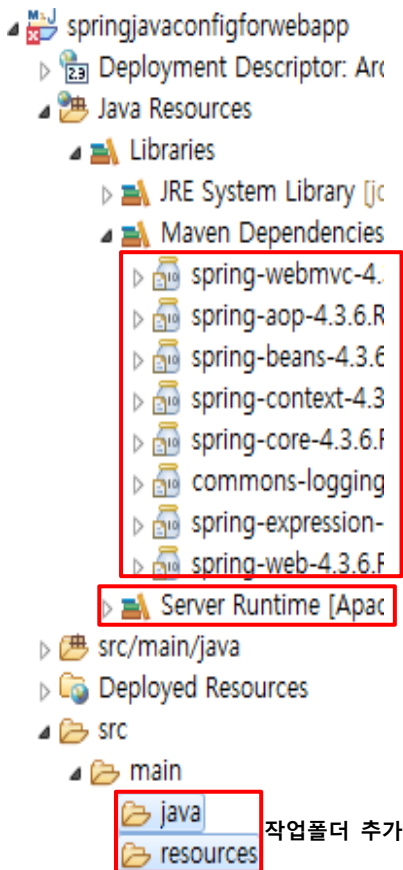
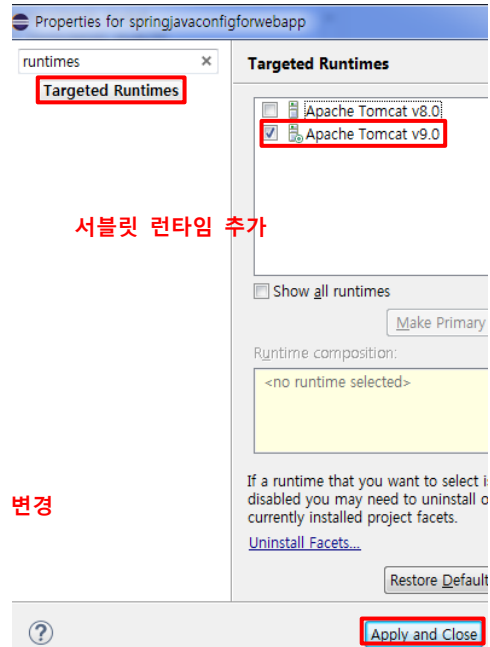
```

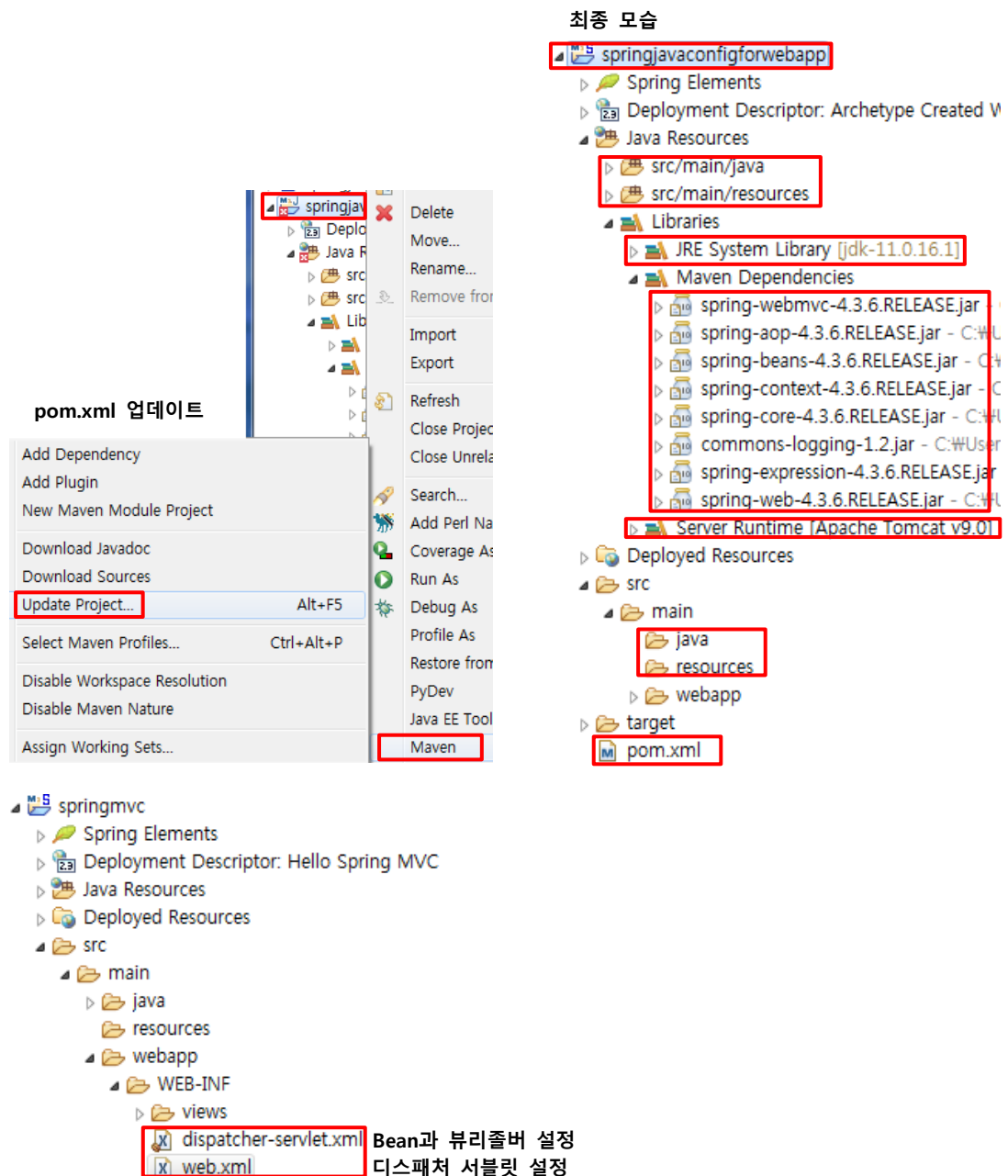
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www
4 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apa
5 <modelVersion>4.0.0</modelVersion>
6
7 <groupId>com</groupId>
8 <artifactId>springjavaconfigforwebapp</artifactId>
9 <version>0.0.1-SNAPSHOT</version>
10 <packaging>war</packaging>
11
12 <name>springjavaconfigforwebapp Maven Webapp</name>
13 <url>http://www.example.com</url>
14
15 <properties>
16 <springframework.version>4.3.6.RELEASE</springframework.version>
17 </properties>
18
19 <dependencies>
20 <dependency>
21 <groupId>org.springframework</groupId>
22 <artifactId>spring-webmvc</artifactId>
23 <version>${springframework.version}</version>
24 </dependency>
25 </dependencies>
26 <build>
27 <pluginManagement>
28 <plugins>
29 <plugin>
30 <groupId>org.apache.maven.plugins</groupId>
31 <artifactId>maven-compiler-plugin</artifactId>
32 <version>3.8.0</version>
33 <configuration>
34 <source>11</source>
35 <target>11</target>
36 </configuration>
37 </plugin>
38 </plugins>
39 </pluginManagement>
40 </build>
41 </project>

```

스프링MVC 모듈 추가

JDK버전을 1.7->11로 변경





3. 마이그레이션 단계

pom.xml	war plugin
dispatcher-servlet.xml	Java Config
web.xml	WebApplicationInitializer

4. pom.xml 업데이트

```
springjavaconfigforwebapp/pom.xml
13 <url>http://www.example.com</url>
14
15 <properties>
16   <springframework.version>4.3.6.RELEASE</springframework.version>
17 </properties>
18
19 <dependencies>
20   <dependency>
21     <groupId>org.springframework</groupId>
22     <artifactId>spring-webmvc</artifactId>
23     <version>${springframework.version}</version>
24   </dependency>
25   <dependency>
26     <groupId>javax.servlet</groupId>
27     <artifactId>javax.servlet-api</artifactId>
28     <version>3.1.0</version>
29     <scope>provided</scope>
30   </dependency>
31 </dependencies>
32
33 <build>
34   <pluginManagement>
35     <plugins>
36       <plugin>
37         <groupId>org.apache.maven.plugins</groupId>
38         <artifactId>maven-compiler-plugin</artifactId>
39         <version>3.8.0</version>
40         <configuration>
41           <source>11</source>
42           <target>11</target>
43         </configuration>
44       </plugin>
45       <plugin>
46         <groupId>org.apache.maven.plugins</groupId>
47         <artifactId>maven-war-plugin</artifactId>
48         <version>2.4</version>
49         <configuration>
50           <warSourceDirectory>src/main/webapp</warSourceDirectory>
51           <warName>springjavaconfigforwebapp</warName>
52           <failOnMissingWebXml>false</failOnMissingWebXml>
53         </configuration>
54       </plugin>
```

Runtime 시점에서 필요없기에 provided로 지정

Webapp/WEB-INF/web.xml이 디폴트로 정해져 있어 없으면 빌드가 실패하는데 이를 false로 지정

5. 자바 설정 클래스 만들기(dispatcher-servlet.xml)

6. 리소스 뷰 리졸버 설정하기(dispatcher-servlet.xml)

7. 디폴트 서블릿 핸들러 설정(web.xml)

8. WebApplicationInitializer 구현

9. 동적으로 디스패처 서블릿 설정

10. 웹 애플리케이션 실행