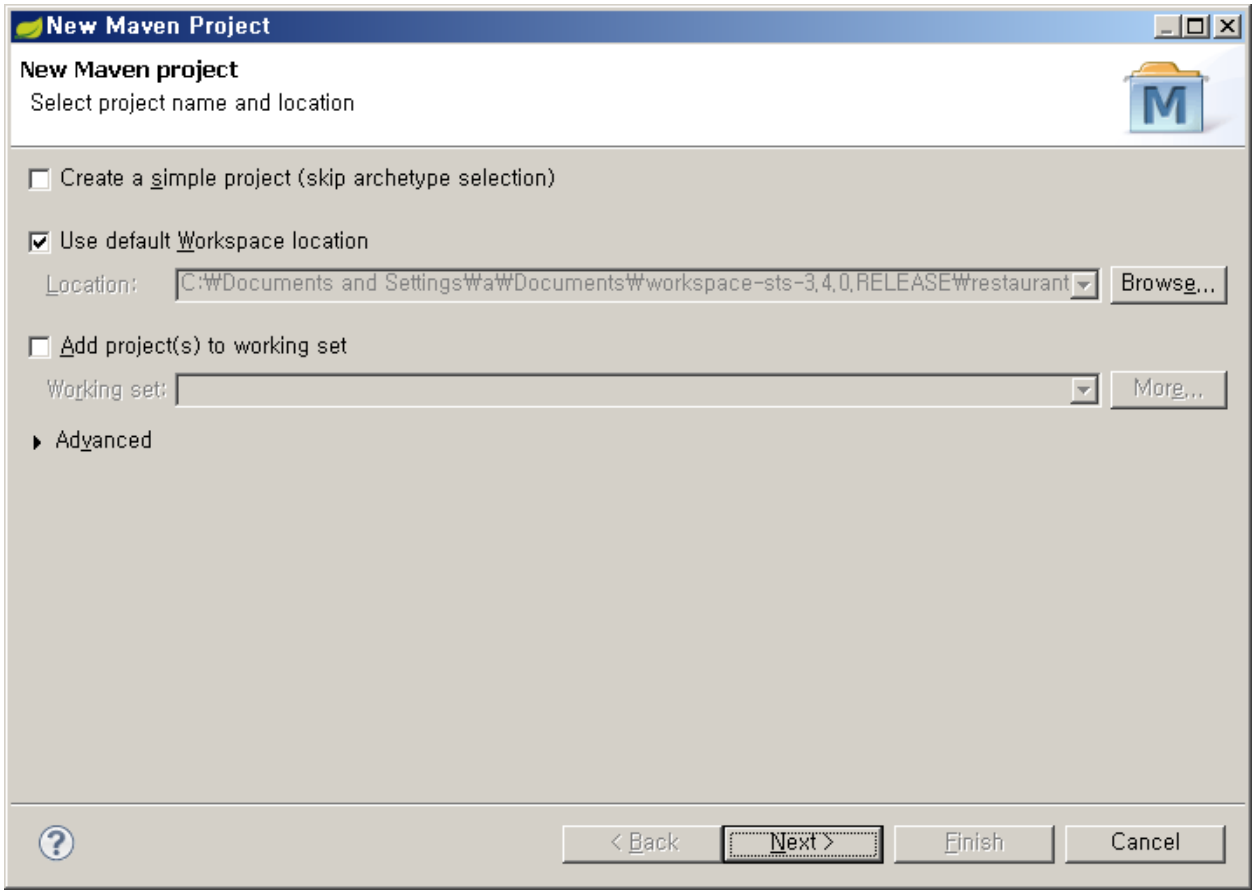
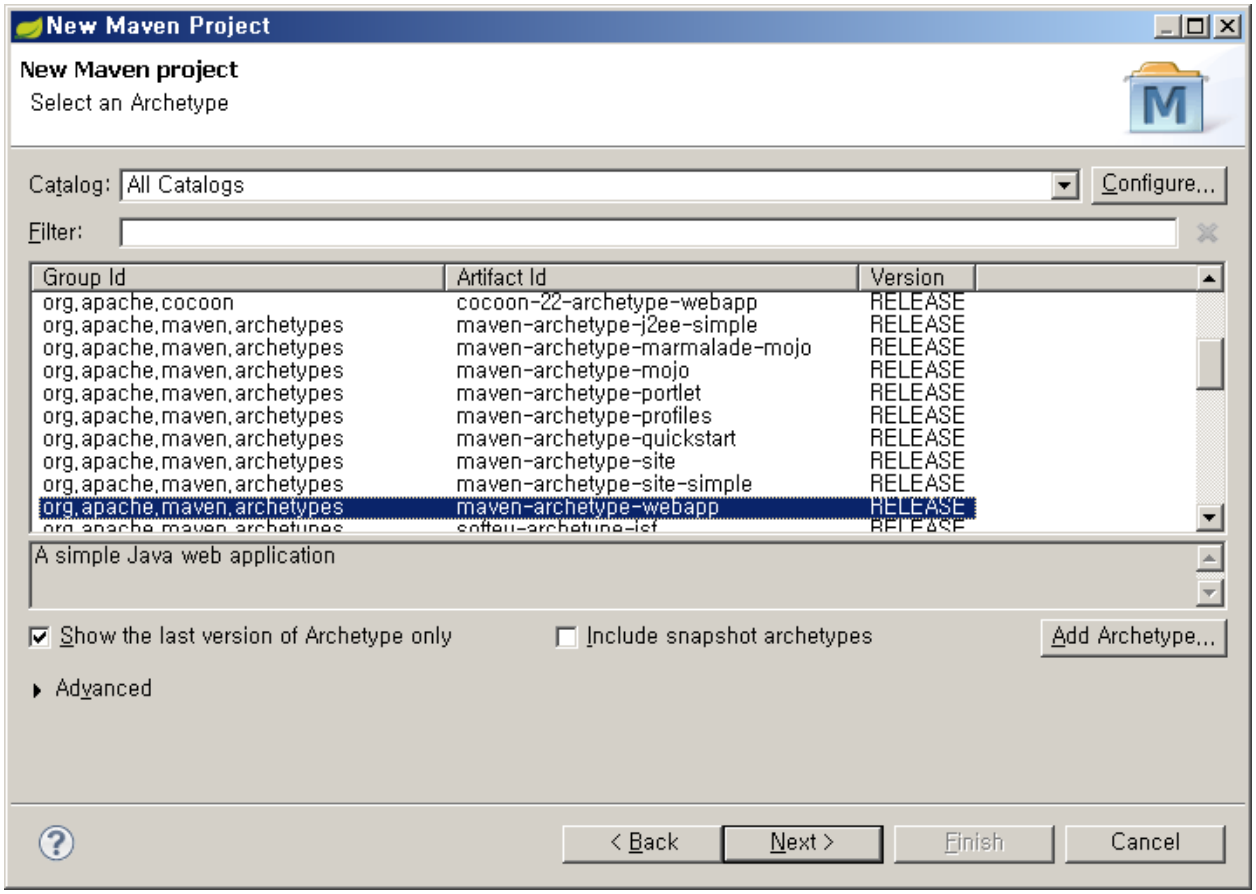
Maven을 사용한 Spring MVC 셋업 방법

1. STEP 1. Maven 프로젝트 만들기

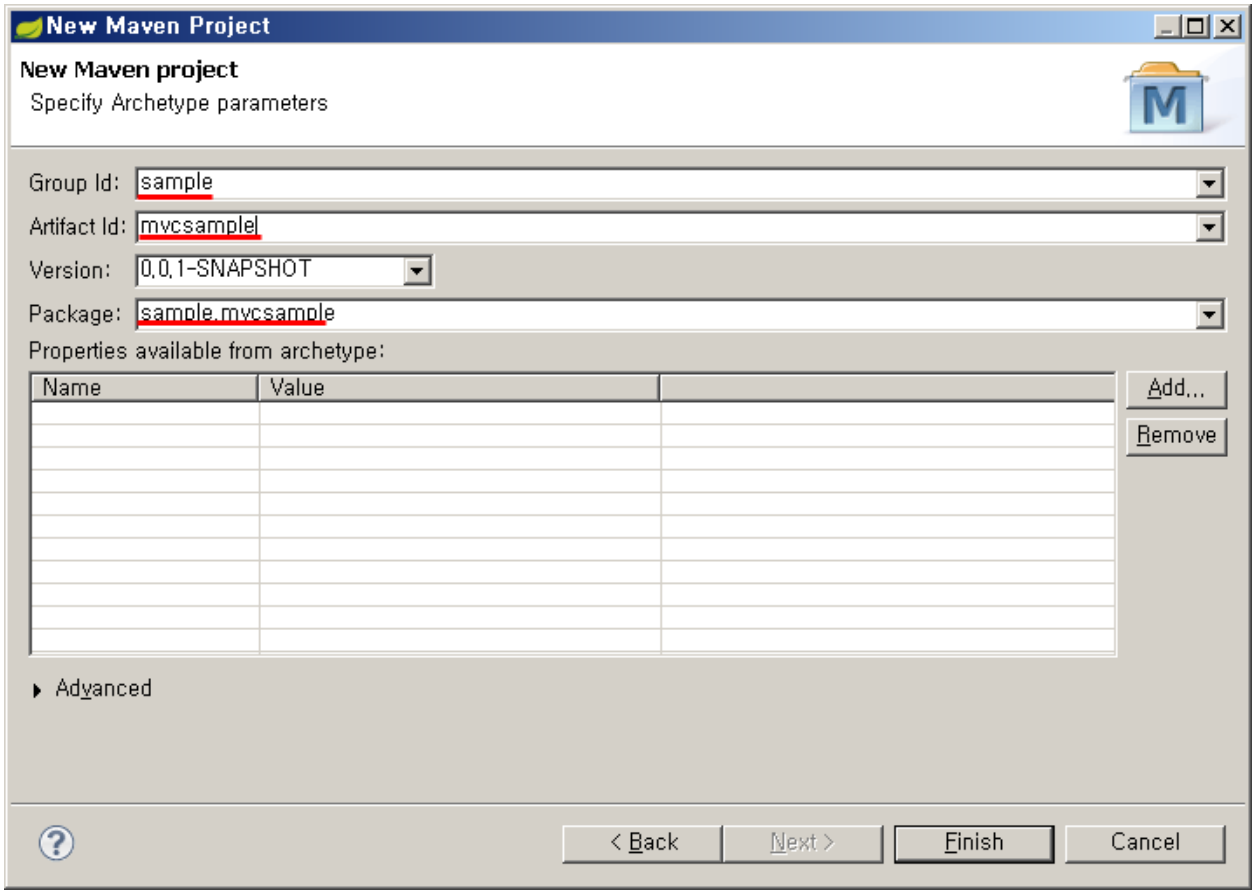
[파일] ⇒ [신규] ⇒ [기타]에서 "Maven 프로젝트"를 선택합니다. 다음에 나타나는 화면에서 그대로 설정에서 "다음"을 클릭합니다 (그림 참조).



1.1.다음에 표시되는 화면에서는 필터 -에 "maven-archetype-webapp"를 입력하고 잠시 기다리면 아래 목록에 "maven-archetype-webapp"라는 아티팩트 ID 항목이 표시되므로 그것을 선택하고 "다음"을 클릭합니다 (그림 참조).

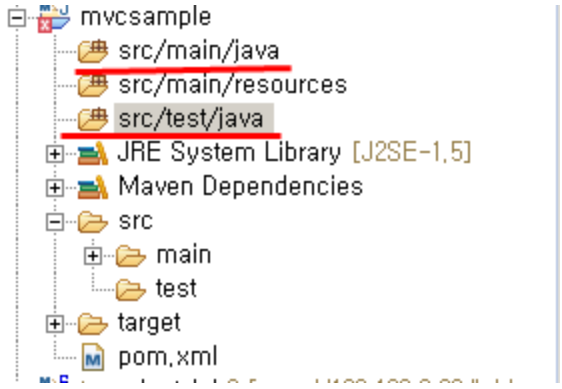


1.2.다음에 나타나는 화면에서 그룹 ID 아티팩트 ID 패키지 이름을 입력하고 "완료"를 클릭합니다 (그림 참조).



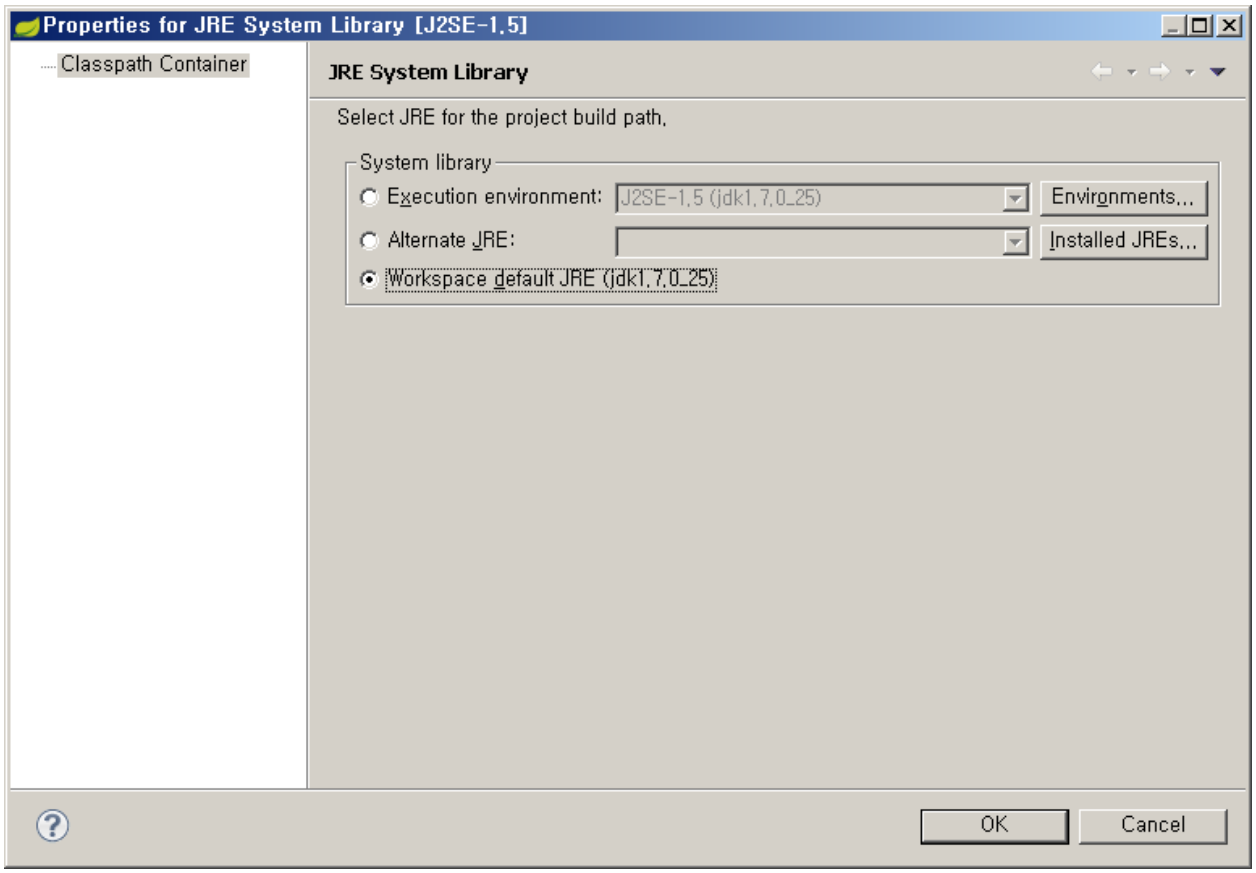
2.STEP 2. 누락 된 소스 폴더 만들기

프로젝트가 생성되면 src / main / resource 만 존재하지만 프로젝트 설정을 보면 소스 폴더로 src / main / java, src / main / resources, src / test / java가 등록되어 있습니다 (버그?) 그래서 수동으로 src / main / java 폴더와 src / test / java 폴더를 만듭니다.



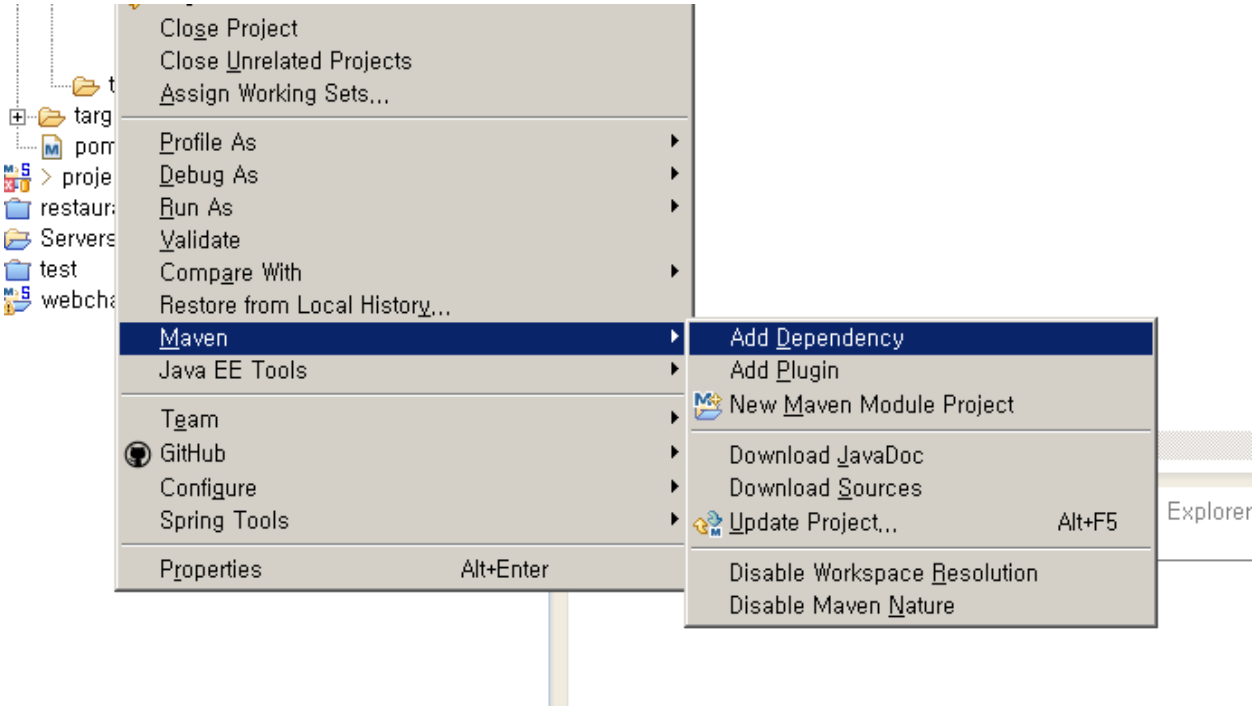
3.STEP 3. JRE 버전 변경

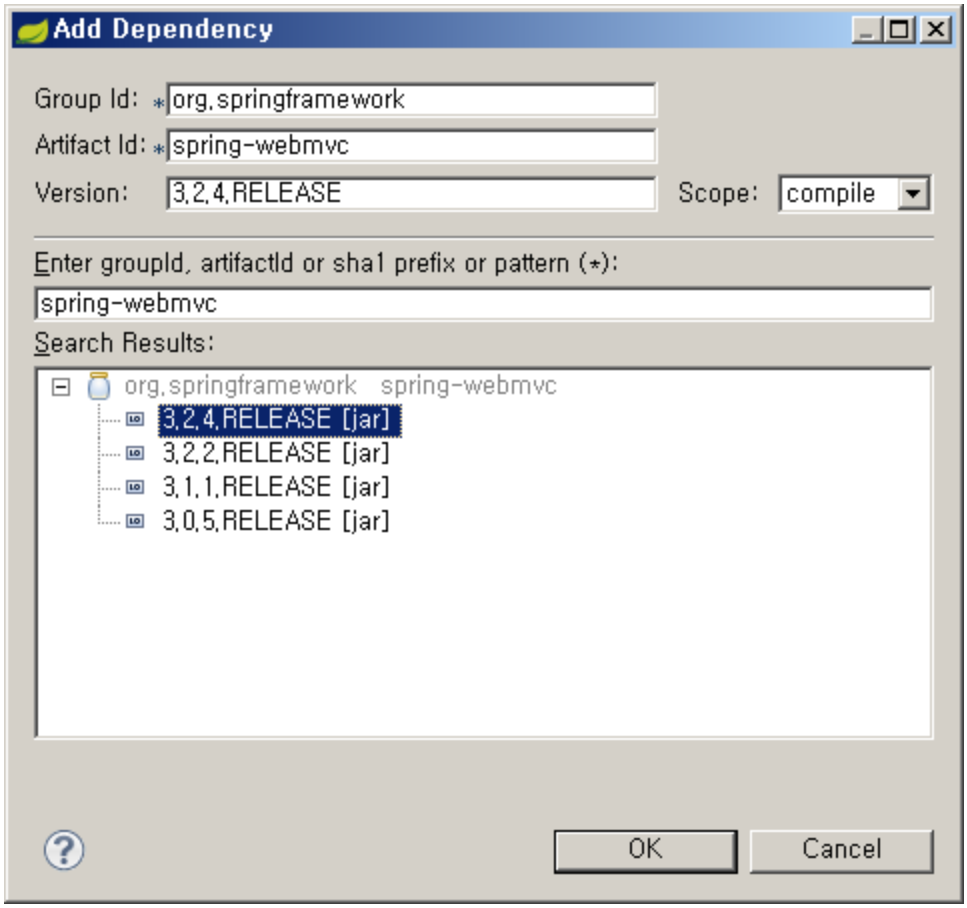
JRE 시스템 라이브러리 버전이 J2SE-1.5이 있기 때문에 JavaSE-1.7 (java7)로 변경합니다.

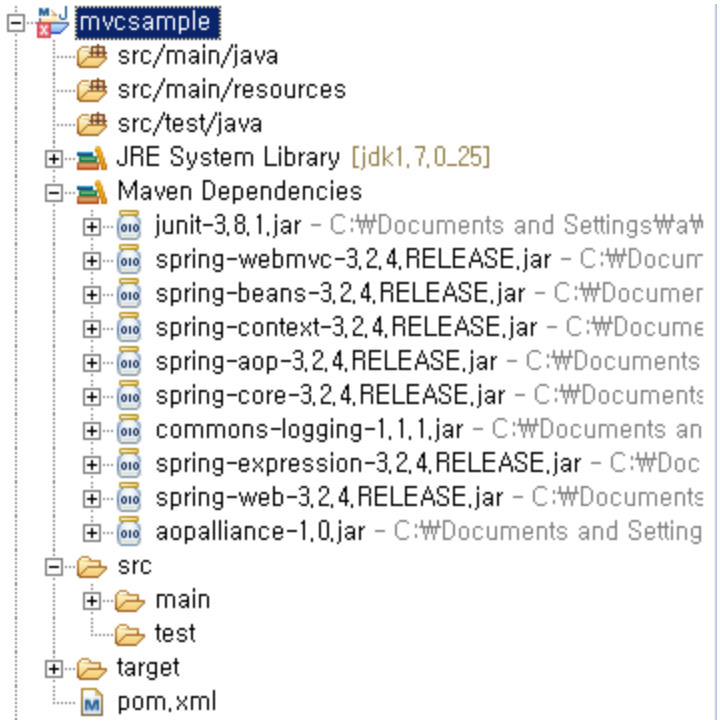


4.STEP 4. Spring Web MVC 라이브러리 추가

Maven 종속성 추가로 org.springframework spring-webmvc를 추가합니다.
프로젝트에서 마우스 오른쪽 클릭 ⇒ [Maven] ⇒ [종속성 추가]로 나타나는 화면에서 "그룹 ID 아티팩트 ID 나 ...」란에 「spring-webmvc"를 입력하면 "org.springframework spring-webmvc "항목이 표시되므로 그것을 선택하고"OK "를 클릭합니다 (그림 참조).


Close Project
Close Unrelated Projects
Assign Working Sets...
Profile As
Debug As
Run As
Validate
Compare With
Restore from Local History...
Maven
Java EE Tools
Team
GitHub
Configure
Spring Tools
Properties Alt+Enter
Add Dependency
Add Plugin
New Maven Module Project
Download JavaDoc
Download Sources
Update Project... Alt+F5
Disable Workspace Resolution
Disable Maven Nature
Explorer


Add Dependency
Group Id: * org.springframework
Artifact Id: * spring-webmvc
Version: 3.2.4.RELEASE
Scope: compile
Enter groupId, artifactId or sha1 prefix or pattern (*):
spring-webmvc
Search Results:
org.springframework spring-webmvc
3.2.4.RELEASE [jar]
3.2.2.RELEASE [jar]
3.1.1.RELEASE [jar]
3.0.5.RELEASE [jar]
OK
Cancel



5.STEP 5. DispatcherServlet 설정

web.xml에 모든 URL 패턴에 org.springframework.web.servlet.DispatcherServlet가 불려 지도록 해주는 서블릿의 설정을 설명합니다.

```
<? xml version = "1.0"encoding = "UTF-8"?>
<web-app
  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
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version = "2.5">
  <display-name> mvcsample </ display-name>
  <! - Creates the Spring Container shared by all Servlets and Filters ->
  <servlet>
    <servlet-name> dispatcher </ servlet-name>
    <servlet-class>
org.springframework.web.servlet.DispatcherServlet </ servlet-class>
    <load-on-startup> 1 </ load-on-startup>
  </ servlet>
  <servlet-mapping>
    <servlet-name> dispatcher </ servlet-name>
    <url-pattern> / </ url-pattern>
  </ servlet-mapping>
</ web-app>
```

6.STEP 6. dispatcher-servlet.xml 만들기

STEP 5.에서 설정 한 'dispatcher'라는 서블릿에 대한 설정 파일을 준비합니다. src / main / webapp / WEB-INF 폴더에 새 XML 파일을 만들고 "dispatcher-servlet.xml"라고 합니다 (그림 참조).

```
<? 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 : context = "http://www.springframework.org/schema/context"
  xmlns : p = "http://www.springframework.org/schema/p"
  xmlns : mvc = "http://www.springframework.org/schema/mvc"
  xsi : schemaLocation = "http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring- context-3.0.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd ">

  <mvc:annotation-driven />

  <context:component-scan base-package="sample.controller" />

  <bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
p:prefix="/WEB-INF/views/" p:suffix=".jsp" />
</ beans>
```