

# Spring Framework

## Framework란

framework가  
Library <

>  
>

소프트웨어를 개발하는데 하나의 기본 구조이다. 기본적인 LOW단(인증, 보안 등)에서 처리하는 부분에 대한 기본구조/골격을 설계해 놓은 상태에서 위에 애플리케이션을 개발할 수 있게 하는 것이다.

예) 제품을 만드는 기본 골격, 골격 안에 부품들을 넣을 수 있는 것을 의미

## Library

다른 프로그램들과 링크되기 위해 존재하는 하나 이상의 서브루틴이나 함수들이 저장된 파일의 모음을 말한다.

예) 제품을 만드는데 결합할 수 있는 부품들

구분	내 용
Spring	엔터프라이즈급 애플리케이션 개발에 필요로하는 경량형 프레임 워크
.NET Framework	마이크로소프트에서 개발한 윈도우 프로그램 개발 및 실행환경
Struts	자바기반의 JSP만을 위한 프레임워크
Django Framework	파이썬 기반의 프레임 워크
AngularJS	자바스크립트 기반 프레임워크



## Spring Framework란?

Enterprise Web

자바 플랫폼을 위한 오픈소스 <애플리케이션> 프레임워크로서 간단히 스프링(Spring)이라고도 불린다.

동적인 웹 사이트를 개발하기 위한 여러 가지 서비스를 제공하고 있으며 대한민국 공공기관의 웹 서비스 개발 시 사용을 권장하고 있는 전자정부 표준프레임워크의 기반 기술로서 쓰이고 있다.

## Spring 공식 사이트

<https://spring.io/>

## Spring의 특징

### I. DI ( Dependancy Injection / 의존성 주입)

설정 파일이나 어노테이션을 통해 객체간 의존 관계를 설정하여 개발자가 직접 의존하는 객체를 생성할 필요가 없다.

### II. Spring AOP ( 관점 지향 프로그래밍)

Database의 트랜잭션처리나 로깅처리와 같이 여러 모듈에서 공통으로 필요로 하는 기능의 경우 해당 기능을 분리하여 관리한다.

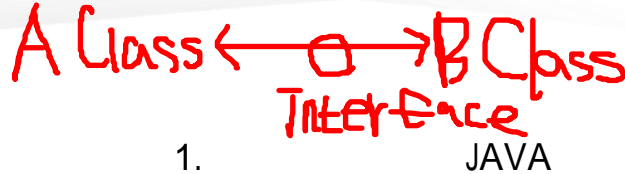
### III. Spring JDBC

Mybatis나 Hibernate 등의 데이터베이스를 처리하는 영속성 프레임워크와 연결할 수 있는 인터페이스를 제공한다.

### IV. Spring MVC

MVC 디자인 패턴을 통해 웹 어플리케이션의 Model, View, Controller 사이의 의존 관계를 DI 컨테이너에서 관리하여 개발자가 아닌 서버가 객체들을 관리하는 웹 어플리케이션을 구축할 수 있다.

# Spring 용어 정리



## POJO(Plain Old Java Object)

-> AClass-BClass

가 A

가

b

자바 객체로 다른 객체와 서로 관계가 설정(상속, 구현)되어 있지 않은 하나의 객체

## IOC(Inversion of Control) = DI

제어의 역행이라는 뜻으로 객체의 관계를 개발자가 아닌 컨테이너가 설정대로 처리하는 것

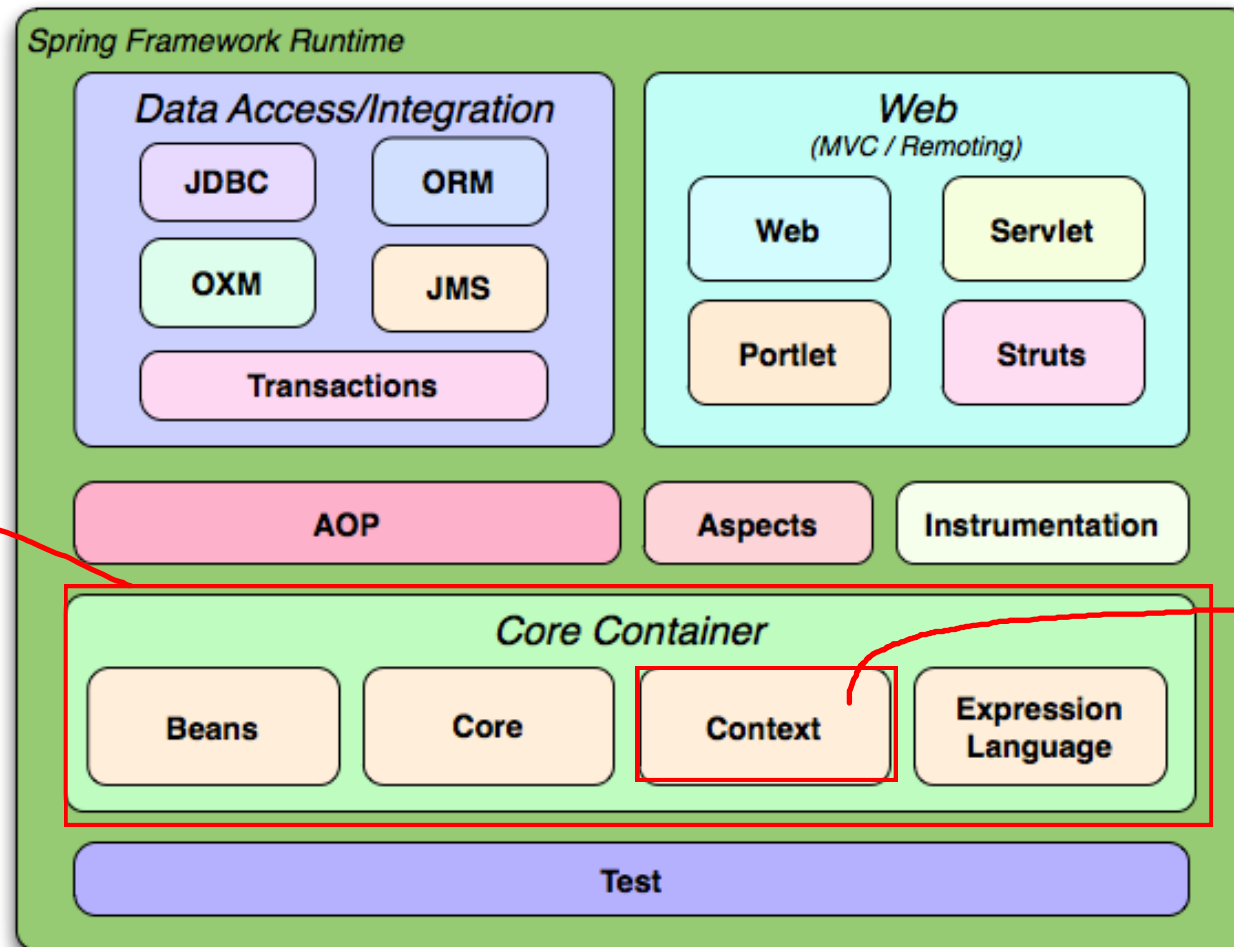
예) Tomcat Spring ~ DI  
EX) Member new Member  
IOC 가 ->

## AOP(Aspect Oriented Programming)

비즈니스 매소드마다 반복되는 공통 로직을 분리하여 관리하는 것



## Spring의 구성 모듈



Spring  
가

## Core Container

Spring의 근간이 되는 IoC(또는 DI) 기능을 지원하는 영역을 담당하고 있다. BeanFactory를 기반으로 Bean 클래스들을 제어할 수 있는 기능을 지원한다.

## Data 접근 계층

JDBC나 데이터 베이스에 연결하는 모듈, Data 트랜잭션에 해당하는 기능을 담당하여 영속성 프레임워크의 연결을 담당한다.



## Web 계층

Spring Framework 에서 Servlet, Struts 등 웹 구현 기술과의 연결점을 Model-View-Controller 구성으로 지원하기 위해 제공하는 모듈 계층이다.

## AOP 계층

Spring에서 각 흐름 간 공통된 코드를 한쪽으로 빼내어 필요한 시점에 해당 코드를 첨부하게 하기 위해 지원하는 계층으로, 별도의 proxy를 두어 동작한다.

## Spring 모듈 정리

모듈명	내 용
spring-beans	스프링 컨테이너를 이용해서 객체를 생성하는 기본기능을 제공
<u>spring-context</u>	객체생성, 라이프 사이클처리, 스키마 확장등의 기능을 제공
<u>spring-aop</u>	AOP기능을 제공한다.
spring-web	REST 클라이언트 데이터 변환 처리, 서블릿 필터, 파일 업로드 지원 등 웹 개발에 필요한 기반 기능을 제공
<u>spring-webmvc</u>	스프링 기반의 MVC프레임워크, 웹 어플리케이션을 개발하는데 필요한 컨트롤러, 뷰 구현을 제공
spring-websocket	스프링 MVC에서 웹 소켓 연동을 처리할 수 있도록 한다.

## Spring 모듈 정리

모듈명	내 용
spring-oxm	XML과 자바 객체간의 매핑을 처리하기 위한 API제공
spring-tx	트랜잭션 처리를 위한 추상 레이어를 제공한다.
spring-jdbc	JDBC프로그래밍을 보다 쉽게 할 수 있는 템플릿 제공
spring-orm	하이버네이트, JPA, Mybatis 등과의 연동을 지원
spring-jms	JMS서버와 메시지를 쉽게 주고 받을 수 있도록 하기 위한 템플릿
spring-context-support	스케줄링, 메일발송, 캐시연동, 벨로시티 등 부가 기능을 제공

## Spring 어플리케이션

Spring 컨테이너

, spring  
spring-context.xml

AbstractApplication 가  
가

AbstractApplication 객체

new GenericXmlApplicationContext()

참조

spring-context.xml

spring컨테이너가 실행해야  
하는 내용을 설정하는 파일

spring-context.xml 1,2,3

1.

트랜잭션관리

관계설정(DI)  
/ 생성

2. class

중복코드 관리  
(AOP)

bean객체

bean객체

공통 메소드 실행



Database

3.

# Spring 동작구조

## Spring 웹

servlet

urlMapping  
/notice/insert

### Servlet 컨테이너

Request

dispatcherServlet

web.xml

urlpattern에 맞는 servlet  
맵핑되어 있는  
dispatcherServlet 생성

### Spring 컨테이너

new XmlWebApplicationContext()

spring-context.xml

스프링 설정xml을 불러옴

ViewResolver

HandlerMapping

Controller  
객체 맵핑

ModelAndView

결과값

Controller 객체

호출View  
경로, 확장자  
가공

호출View  
경로, 확장자  
가공

호출View  
경로, 확장자  
가공

## XML파일

Spring 컨테이너 구동시 한 개의 spring 환경설정된 xml파일을 불러오는데 이 파일에 bean, aop, transaction 등 여러 사항을 다 작성하여 구동하는 방식

```
<?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"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.2.xsd">

  <!-- controllerMapping -->

  <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
      <props>
        <prop key="/login.do">login</prop>
        <prop key="/board.do">board</prop>
      </props>
    </property>
  </bean>
  <bean id="login" class="com.kh.mvc2.user.controller.LoginController"></bean>
  <bean id="board" class="com.kh.mvc2.board.controller.BoardController"></bean>

  <!-- viewResolver -->

  <bean id="viewResolver"
    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/board/"></property>
    <property name="suffix" value=".jsp"></property>
  </bean>
```

## @Annotation

xml파일에는 구동시킬 필수요소만 작성하고 소스코드에 Annotation으로 표시하여 구동하는 방식

```
<?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"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.2.xsd">
  <!-- 어노테이션 적용 -->
  <context:component-scan base-package="com.kh.mvc2"></context:component-scan>
```

xml 파일

```
package com.kh.mvc2.board.controller;

import java.util.List;

@Controller
public class BoardController{
    @RequestMapping(value="/board.do", method=RequestMethod.GET)
    public ModelAndView getListGet(HttpServletRequest request, HttpServletResponse response) {
        System.out.println("get");
        BoardVo board=new BoardVo();
        List<BoardVo> list=new BoardDAO().getList();

        ModelAndView mv=new ModelAndView();
        mv.addObject("boards",list);
        mv.setViewName("/WEB-INF/board/boardlist.jsp");

        return mv;
    }
    @RequestMapping(value="/board.do", method=RequestMethod.POST)
    public ModelAndView getListPost(HttpServletRequest request, HttpServletResponse response) {
        System.out.println("Post");
        BoardVo board=new BoardVo();
        List<BoardVo> list=new BoardDAO().getList();

        ModelAndView mv=new ModelAndView();
        mv.addObject("boards",list);
        mv.setViewName("/WEB-INF/board/boardlist.jsp");

        return mv;
    }
}
```

소스코드

# STS 설치

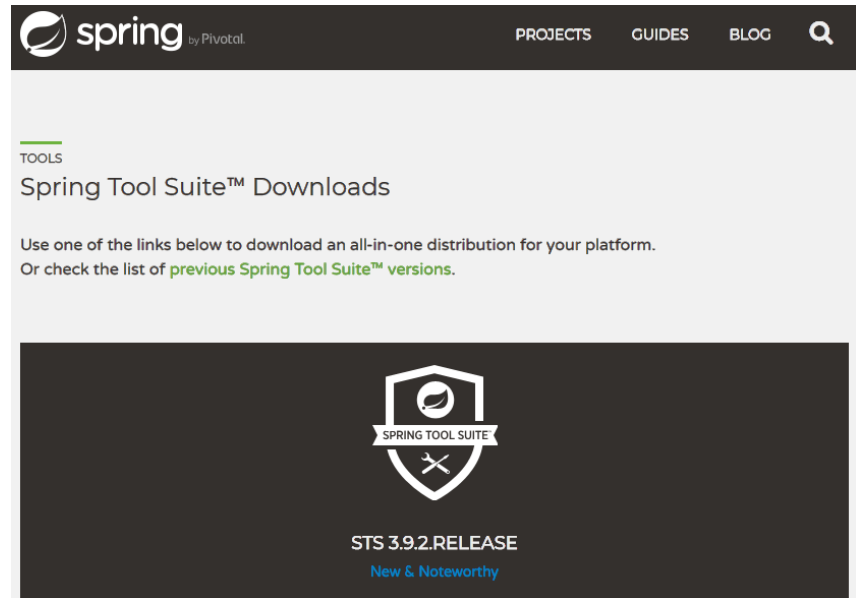


## STS란?

Spring Tool Suite의 약자로, Spring Framework를 사용하기 위한 개발 툴을 말한다. 일반적으로는 별도의 설치 도구를 통해 설치하여 사용하나, 이클립스IDE에서 제공하는 STS plug-in을 통해 간단히 설치할 수 있다.

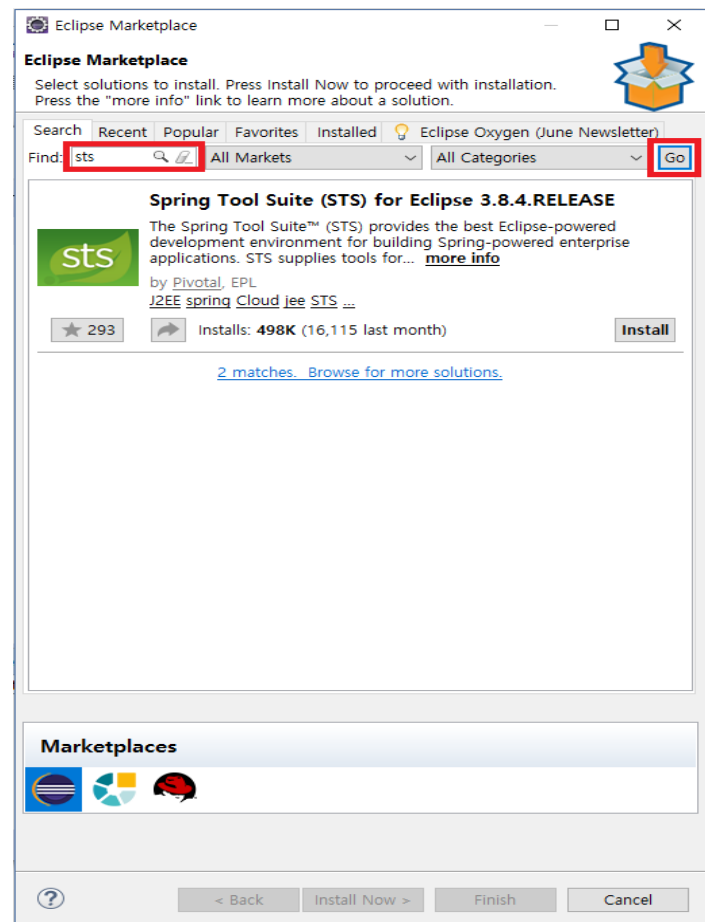
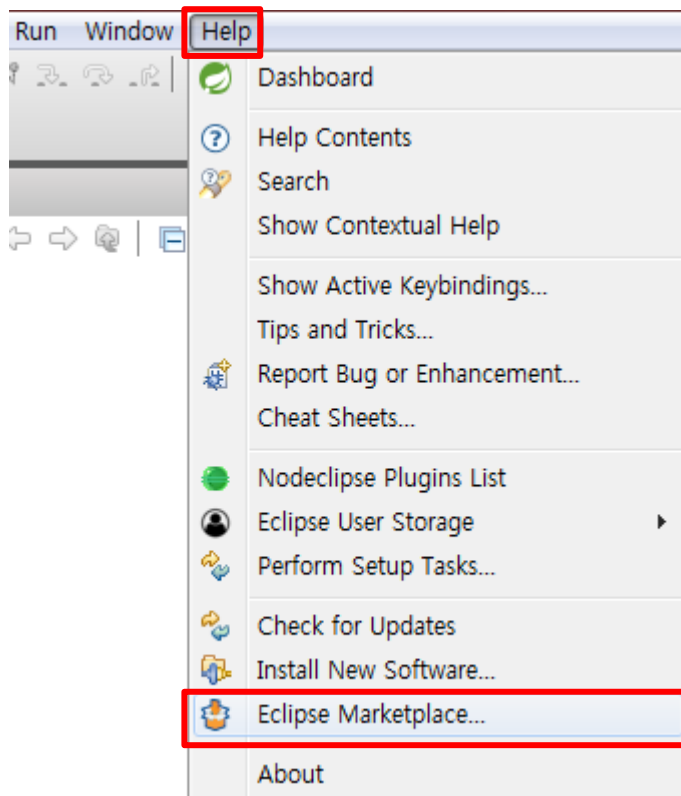
## STS 공식 설치 사이트

<https://spring.io/tools/sts/all>



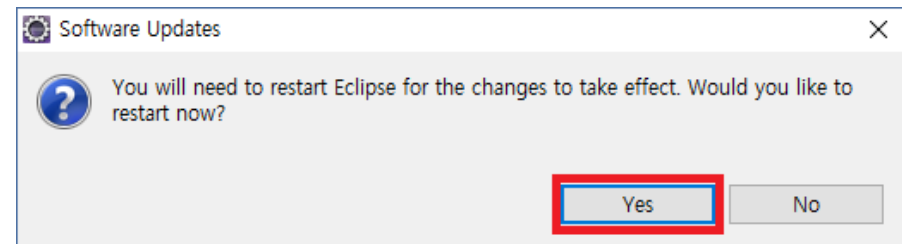
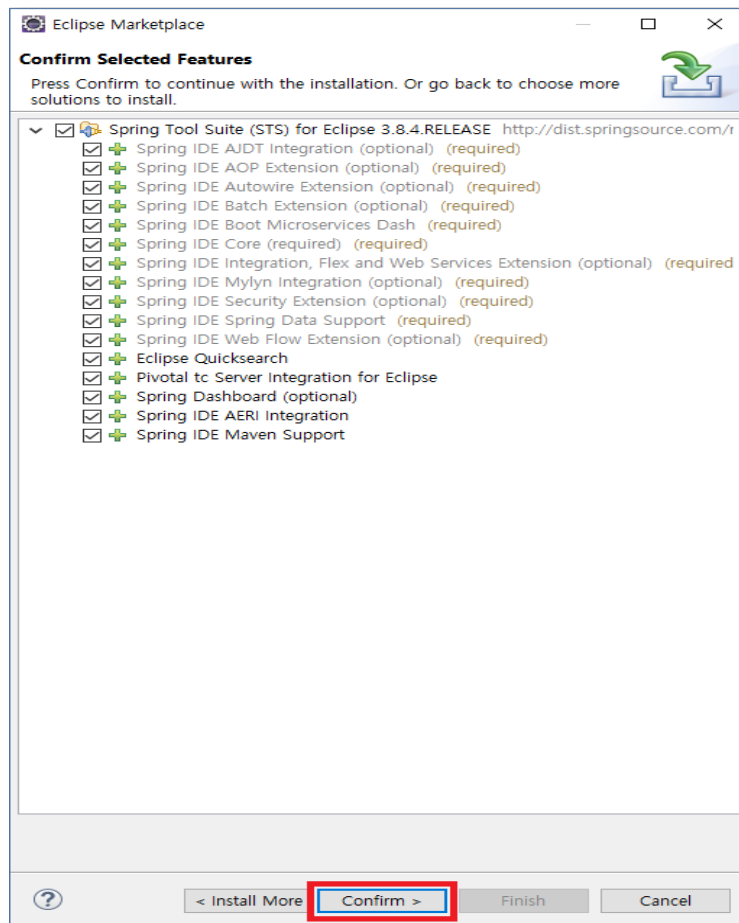
## Eclipse STS 설치

이클립스 메뉴에서 [ Help ] - [ Eclipse Marketplace ]를 클릭하고  
Find : sts 입력하고 [ Go ] 클릭하고 Spring Tool Suite 를 설치한다.



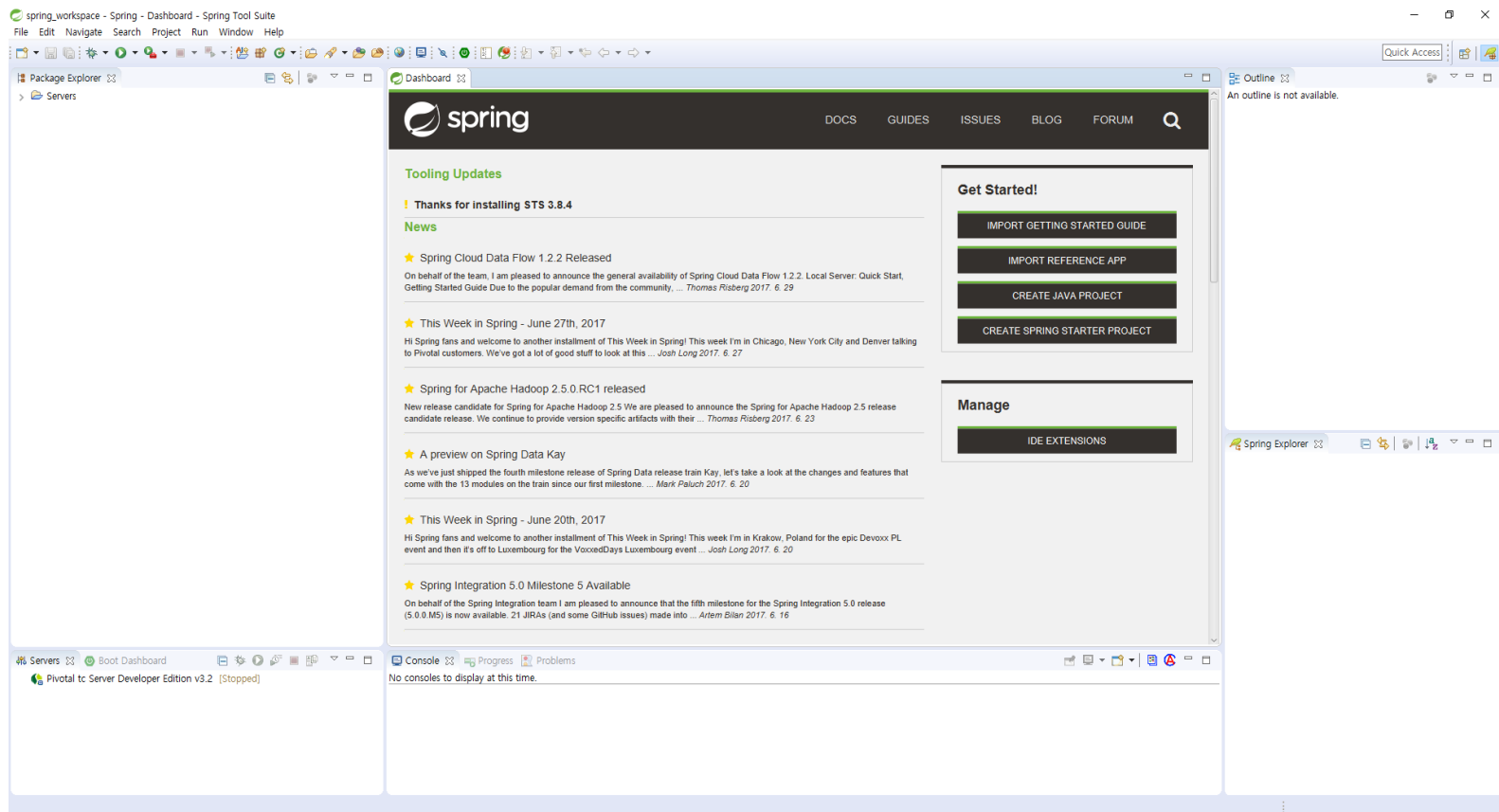
## Eclipse STS 설치

설치 시 나오는 항목을 모두 선택하고, 설치가 완료되면 이클립스를 재실행한다.



## Eclipse STS 설치

재 실행 시 다음과 같은 Spring Welcome page가 첫 화면에 보인다면 설치가 정상적으로 이루어진 것이다.



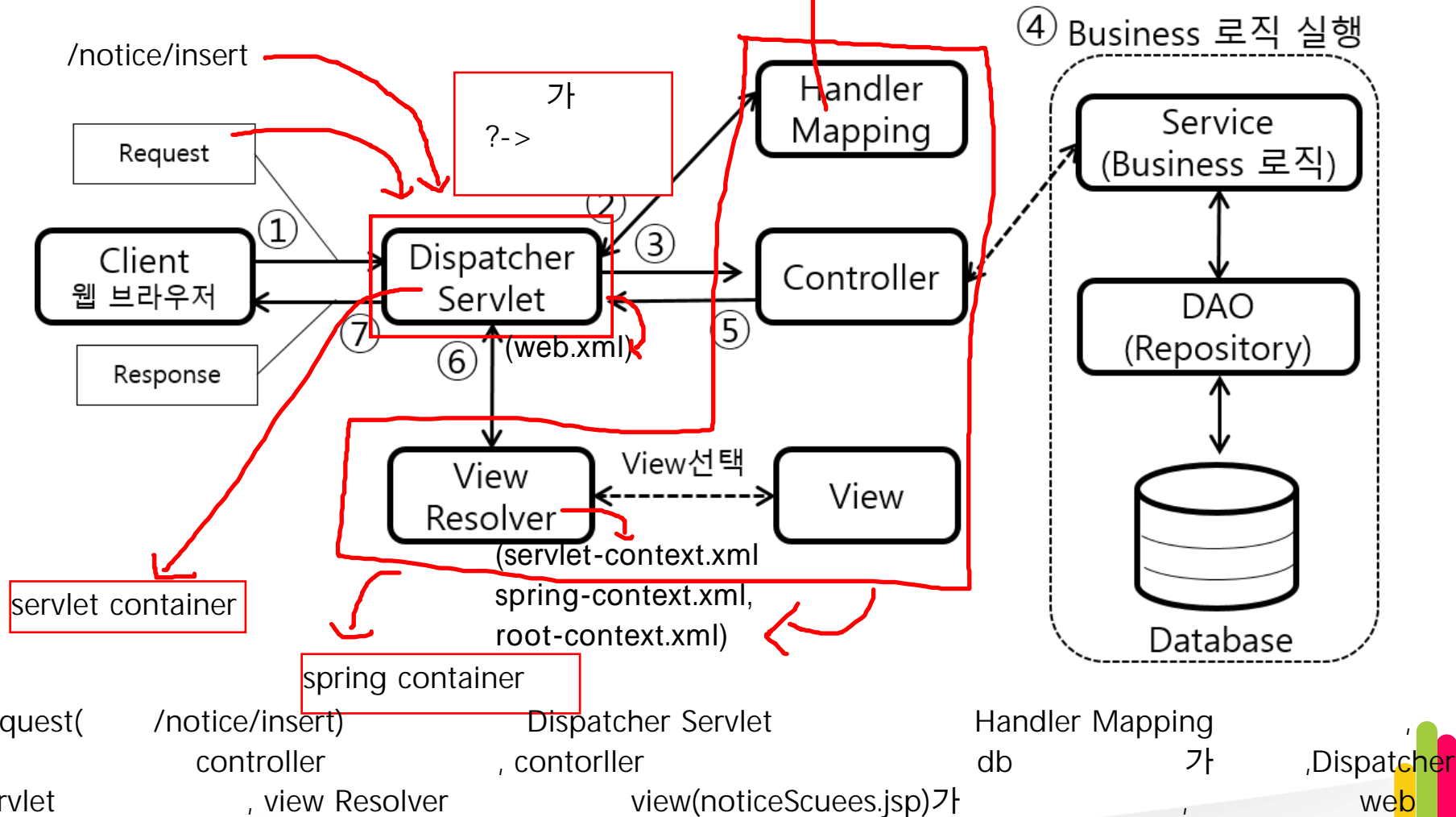
# Spring MVC

## Spring MVC

Spring Framework에서는 클라이언트의 화면을 표현하기 위한 View와 서비스를 수행하기 위한 개발 로직 부분(Model)을 나누는 MVC 2 패턴을 지원한다. 또한 Model, View, Controller 사이의 의존 관계를 DI 컨테이너에서 관리하여 유연한 웹 어플리케이션을 쉽게 구현 및 개발할 수 있다.

## Spring MVC 요청 처리 과정

어노테이션 @RequestMapping



## Spring MVC 구성 요소

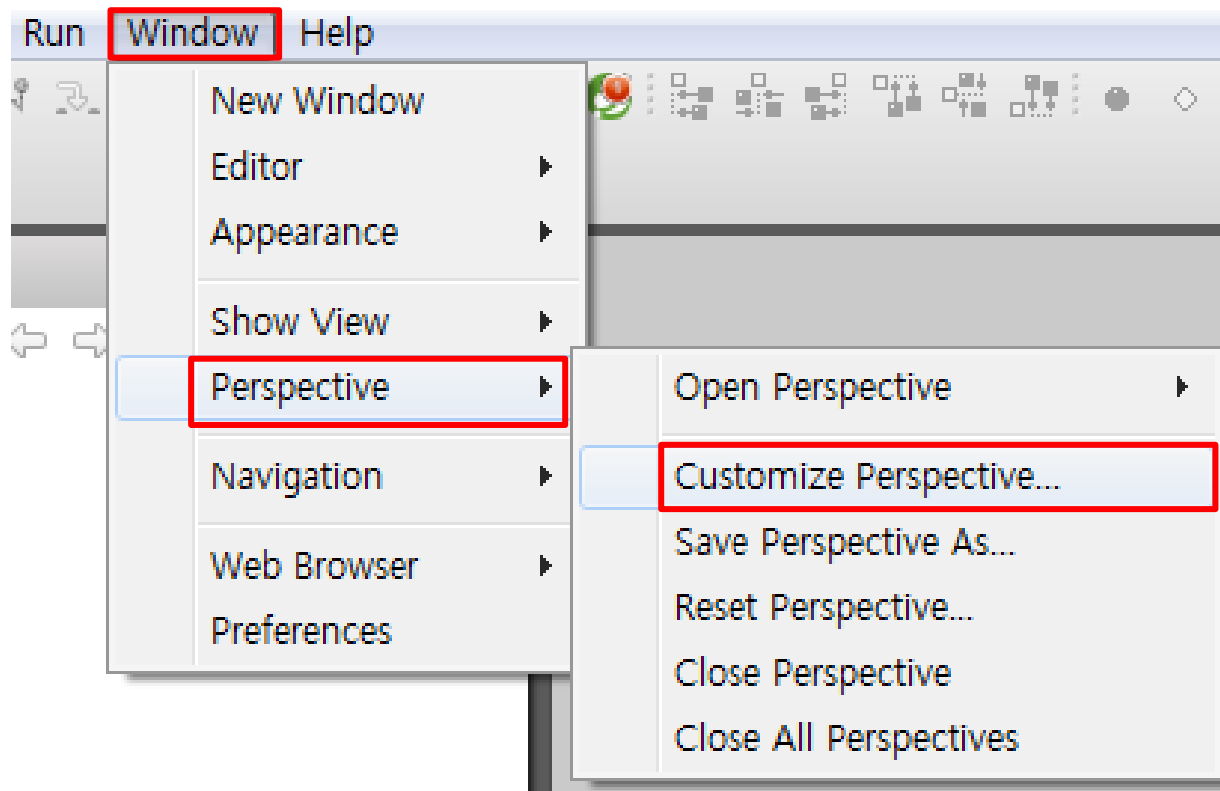
구성 요소	설명
* Dispatcher Servlet	클라이언트의 요청을 전달받음 요청에 맞는 컨트롤러가 리턴한 결과값을 View에 전달하여 알맞은 응답을 생성
Handler Mapping	클라이언트의 요청 URL을 어떤 컨트롤러가 처리할지 결정
Controller	클라이언트의 요청을 처리한 뒤, 결과를 DispatcherServlet에게 리턴
Model AndView	컨트롤러가 처리한 결과 정보 및 뷰 선택에 필요한 정보를 담음
ViewResolver	컨트롤러의 처리 결과를 생성할 View를 결정
View	컨트롤러의 처리 결과 화면을 생성, JSP나 Velocity 템플릿 파일 등을 View로 사용

화면



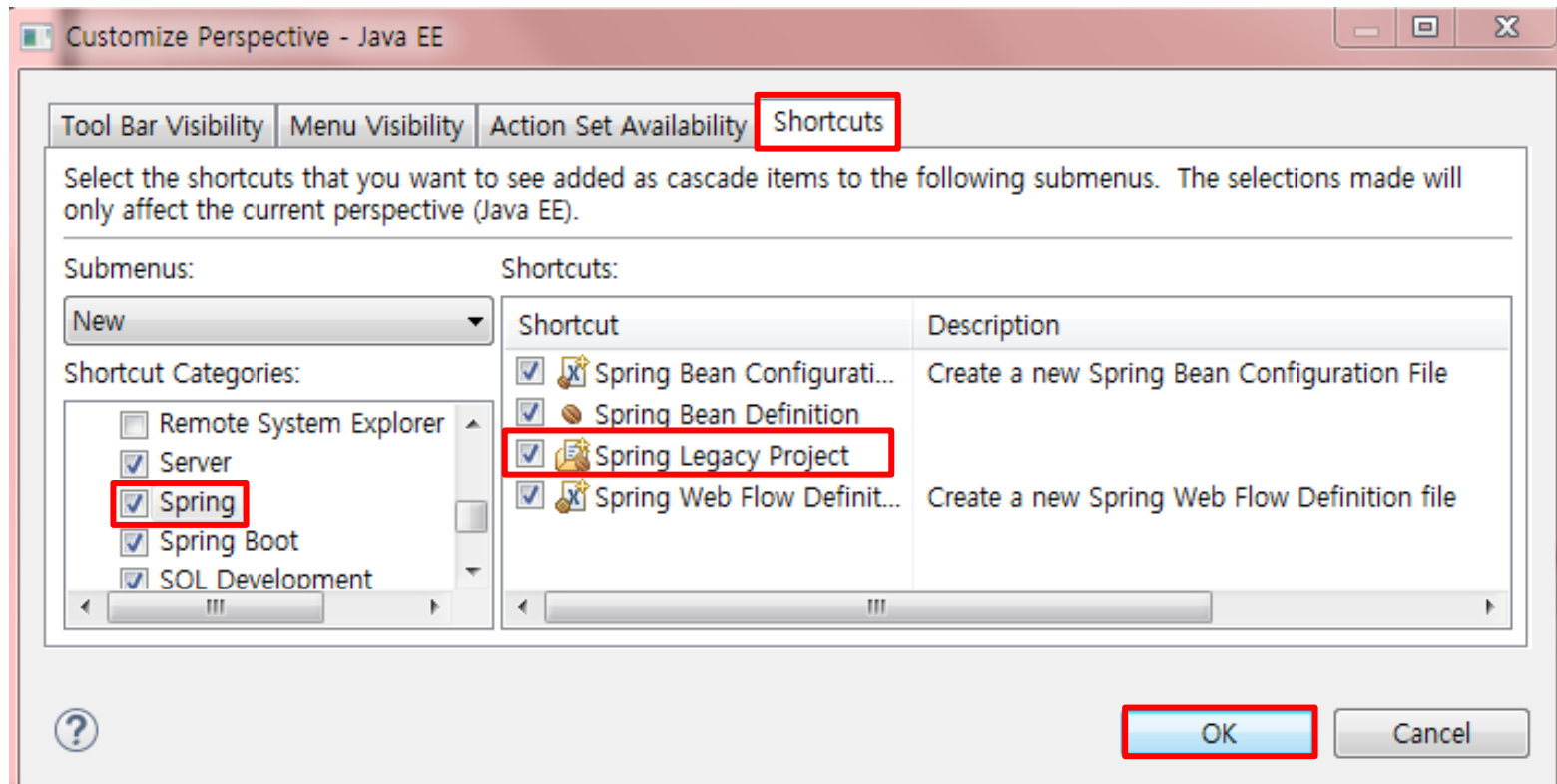
## Spring MVC 프로젝트 만들기

이클립스 [Window] - [Perspective] - [Customize Perspective]를 클릭



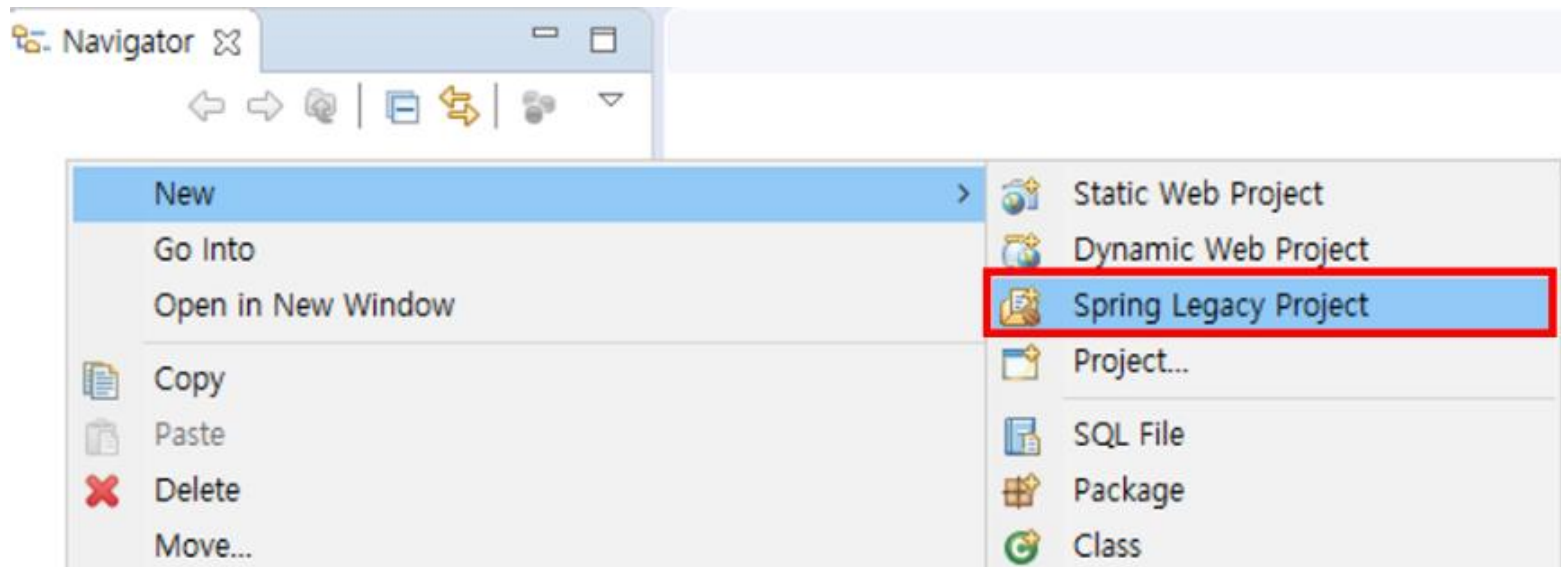
## Spring MVC 프로젝트 만들기

Shortcuts 탭에서 Spring Legacy Project를 선택하고 OK 클릭



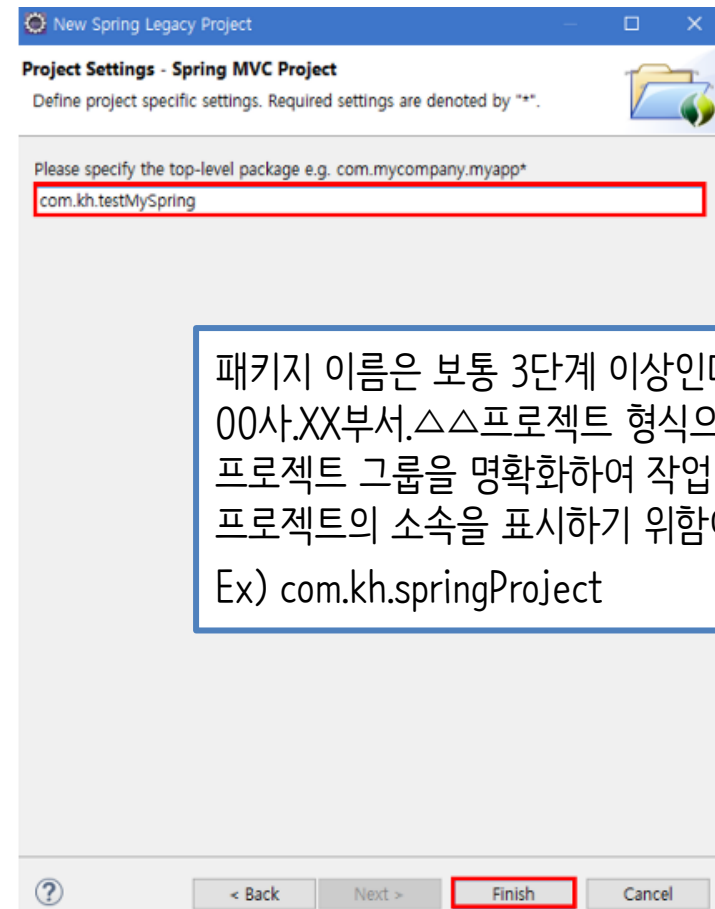
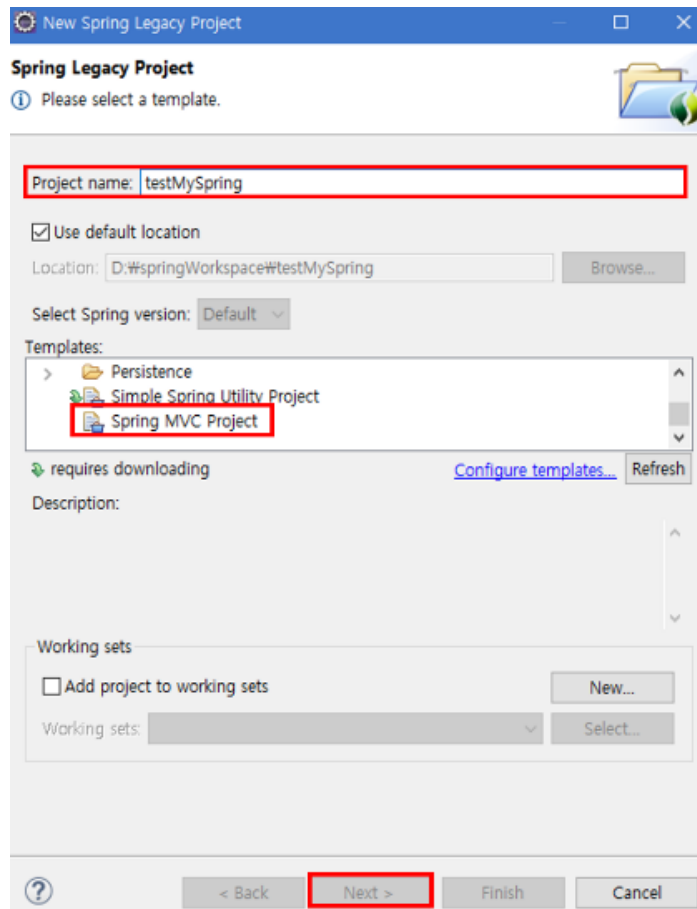
## Spring MVC 프로젝트 만들기

Project Explorer 창에서 [New] - [Spring Legacy Project] 선택



## Spring MVC 프로젝트 만들기

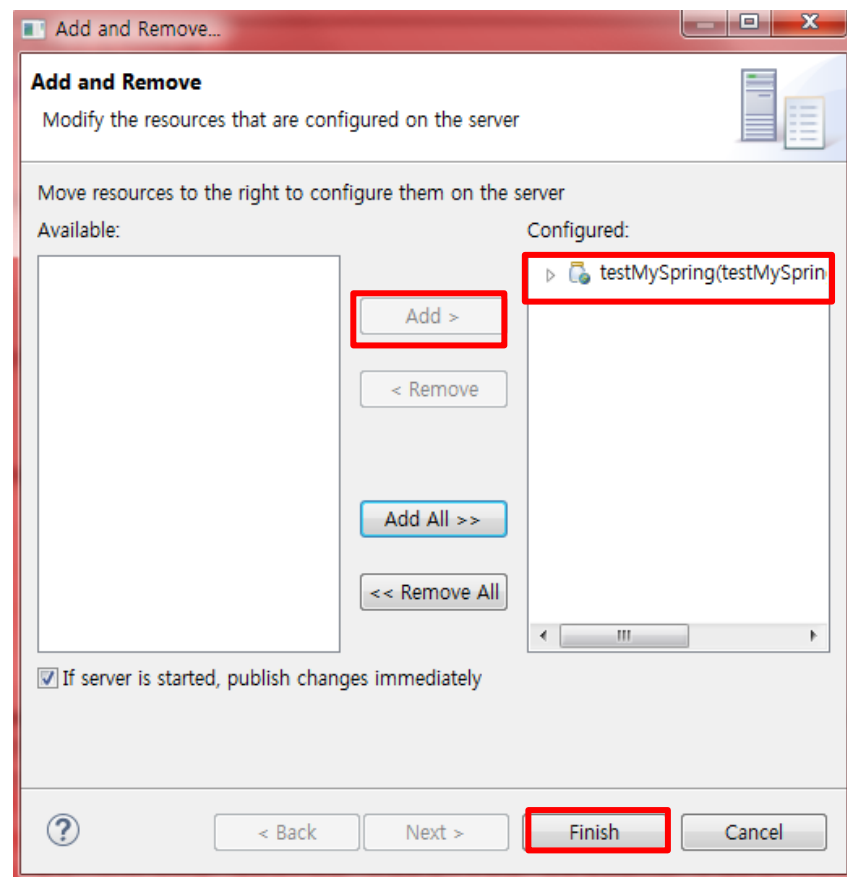
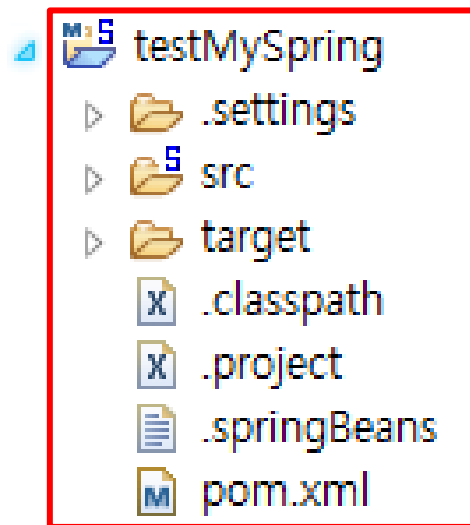
프로젝트 이름을 정하고, Template에서 Spring MVC를 선택하고 [Next],  
패키지 생성 창에서 'com.kh.프로젝트 명' 으로 패키지 생성



패키지 이름은 보통 3단계 이상인데,  
00사.XX부서.△△프로젝트 형식으로  
프로젝트 그룹을 명확화하여 작업 중인  
프로젝트의 소속을 표시하기 위함이다.  
Ex) com.kh.springProject

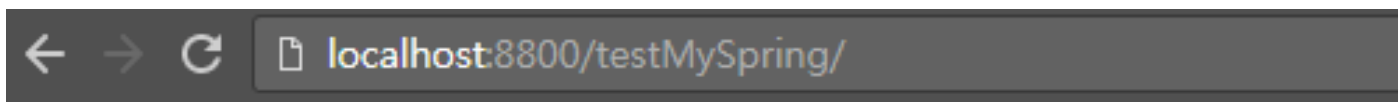
## Spring MVC 프로젝트 만들기

생성이 완료된 프로젝트를 서버에 추가하고 프로젝트 명으로 접속한다.



## Spring MVC 프로젝트 만들기

생성이 완료된 프로젝트를 서버에 추가하고 프로젝트 명으로 접속한다.



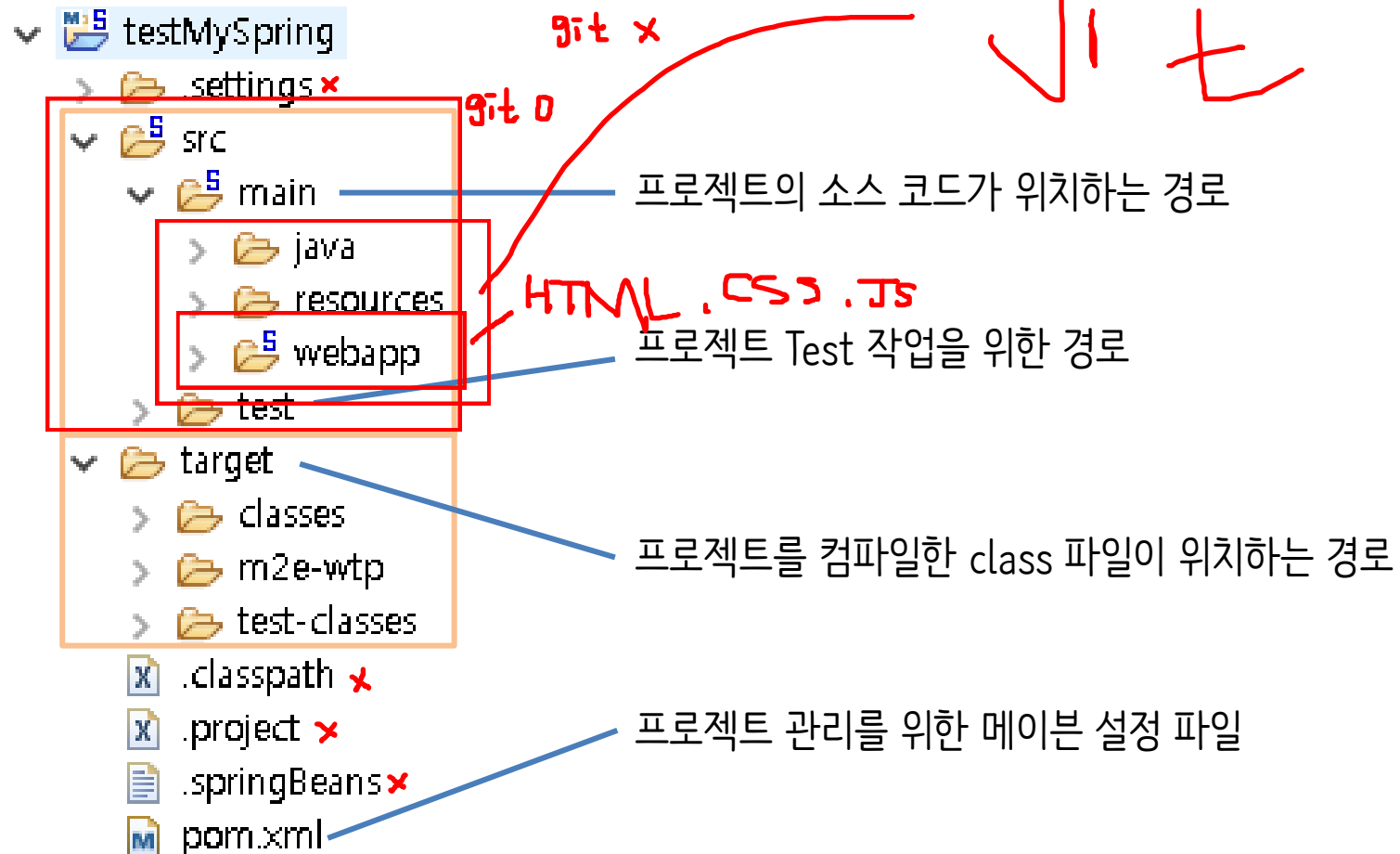
# Hello world!

The time on the server is 2018년 1월 29일 (월) 오전 1시 31분 44초.

# Spring 프로젝트 구조

## Spring 폴더 구조

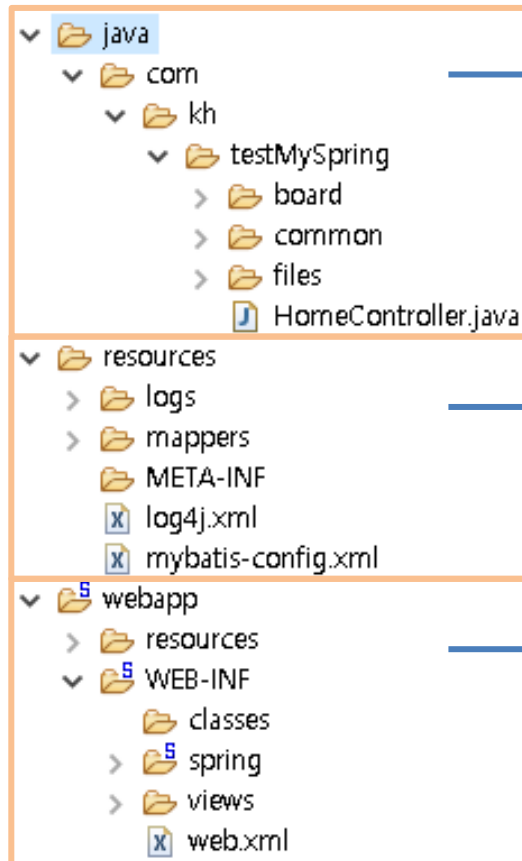
Spring Framework는 다음과 같은 폴더 구조를 가진다.





## main folder

main 폴더는 다음과 같은 구성을 가진다.



java

java

우리가 작성하는 java 파일의 위치

resources

.xml

프로젝트 설정에 필요한 xml 등의 설정파일들

webapp

html.css.js

사용자 화면에 표시할 view 관련 파일들과  
웹 컨테이너 설정에 필요한 xml 파일들

## webapp folder

웹 상의 루트 역할인 webapp 폴더는 다음과 같은 구성을 가진다.

