

# chapter01

## spring 시작하기

1. **spring** 개요
2. 프로젝트 만들기 (hello spring)
3. 스프링의 DispatcherServlet 과 MVC

## ■ 스프링이란?

자바 엔터프라이즈 개발을 편하게 해주는  
오픈소스 경량급 애플리케이션 프레임워크

# 01 스프링이란?

## ■ 자바 엔터프라이즈 개발

- 기업대상 애플리케이션 개발

  - ex) 은행, 네이버, 물류회사, 병원, .... future business

- 환경과 조건

  - C/S ( network )

  - 웹환경

  - 보안

  - 분산환경 ( 분산객체, 자원관리, 컴포너트 자원....)

- J2EE ( servlet, jsp, JDBC, JNDI, JavaMail, JTA, RMI )

## ■ 프레임워크

- 프로그램을 만드는 데 아주 기본이 되는 골격 코드
- 반제품
- 완전한 애플리케이션 소프트웨어가 아니다.
- 문제영역을 해결하기 위한 잘 설계된 재사용 가능한 모듈
- 확장하여 비즈니스 요구사항에 맞는 완전한 애플리케이션으로 완성이 요구된다.
- 웹 애플리케이션 프레임워크
  - Struts, WebWork, Spring MVC
- 데이터베이스 관련 프레임워크
  - iBatis, Hibernate, Spring DAO

## ■ 어플리케이션 프레임워크

- 특정계층, 기술, 업무에 국한되지 않은 애플리케이션 전영역을 포괄
- 개발 전 과정을 빠르고 편리하며 효율적으로 진행하는 것을 목표
- 웹 MVC 프레임워크, JDBC/ORM 프레임워크
- IoC / DI 프레임워크, AOP 툴
- 스프링의 목적은 핵심 기술에 담긴 프로그래밍 모델을 일관되게 적용하여 엔터프라이즈 애플리케이션 전 계층과 전 영역에 전력과 기능을제공, 애플리케이션을 편리하게 개발하게 해주는 애플리케이션 프레임워크로 사용되는 것

## 01 스프링이란?

### ■ 경량급

- 20여개의 모듈과 수십만 라인의 복잡하고 방대한 규모
- **불필요하게 무겁지 않다.** ( EJB와 비교 )
- 코드는 단순하고 개발과정은 편리
- 고급 기능을 세련된 방식으로 적용
- 군더더기 없이 깔끔한 기술을 가진 "경량급" 프레임워크
- 비슷한 기술 수준에서 훨씬 빠르고 간편하게 작성이 가능

# 01 스프링이란?

## ■ POJO 프로그래밍

- 스프링의 정수는 엔터프라이즈 서비스 기능을 POJO에 제공하는 것
- 스프링의 핵심 POJO
  - Plain Old Java Object
    - 자바 언어와 꼭 필요한 API외에는 특정 규약에 종속되지 않는다.
    - 특정 환경에 종속되지 않는다. ( 기술과 비즈니스 분리 )
    - 진정한 POJO는 객체지향 원리에 충실하면서 환경과 기술에 종속되지 않고 필요에 따라 재활용될 수 있는 방식으로 설계된 오브젝트

## 01 스프링이란?

### ■ IoC (제어역전) / DI (의존관계 주입)

- 스프링의 가장 기본이 되는 기술이자 스프링 핵심 개발 원칙
- bean : 스프링이 제어권을 가지고 직접 만들고 관계를 부여하는 오브젝트
- 스프링 빈은 스프링 컨테이너가 생성과 관계 설정 등을 제어
- IoC(DI) Container = Bean Factory = Application Context



## ■ AOP ( Aspect Oriented Programming )

- 관점 지향 프로그래밍
- OOP를 더욱 더 OOP답게 해주는 ( 더욱 더 완벽하게 해 주는) 기술
- 관심의 분리 ( Seperation of Concern )
- 횡단 관심( Crosscutting Concern )과 핵심관심( Core Concern )
- 핵심관심 모듈과 횡단 관심 모듈이 긴밀하게 결합 ( 핵심 모듈이 필요한 시점에..)
- OOP 문제점 : 중복코드, 지저분한 코드, 생산성 저하, 재활용성의 문제점
- 필요한 시점에 횡단관심 모듈을 삽입하여 동작하게 하는 기술.
- EJB AOP, JDK Dynamic Proxy, AspectJ, Spring AOP

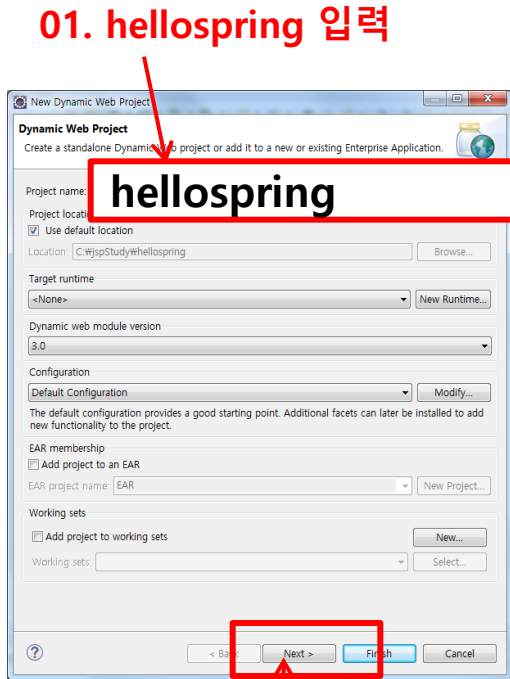
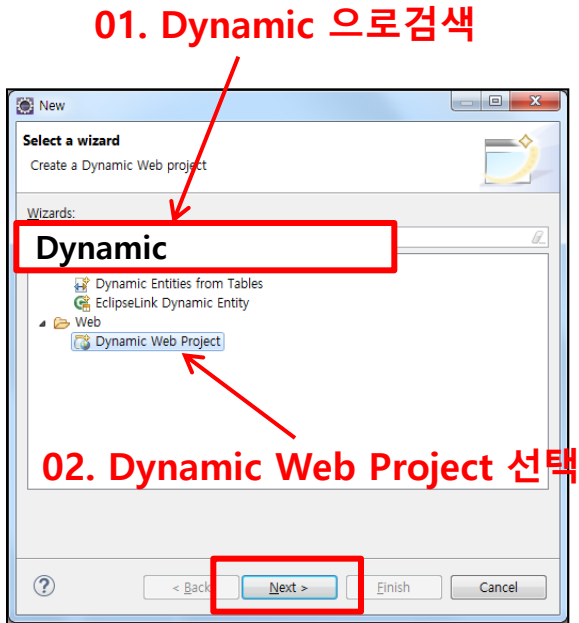
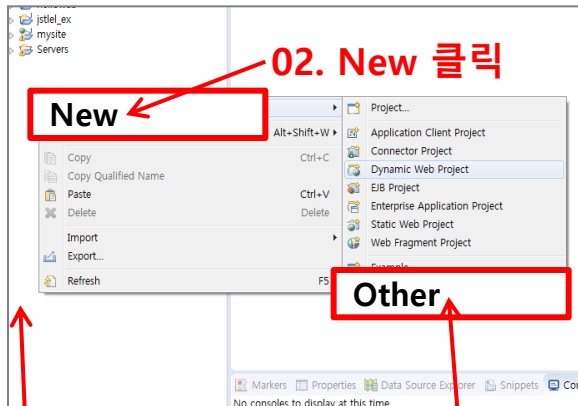
# chapter01

## spring 시작하기

1. spring 개요
2. **프로젝트 만들기 (hello spring)**
3. 스프링의 DispatcherServlet 과 MVC

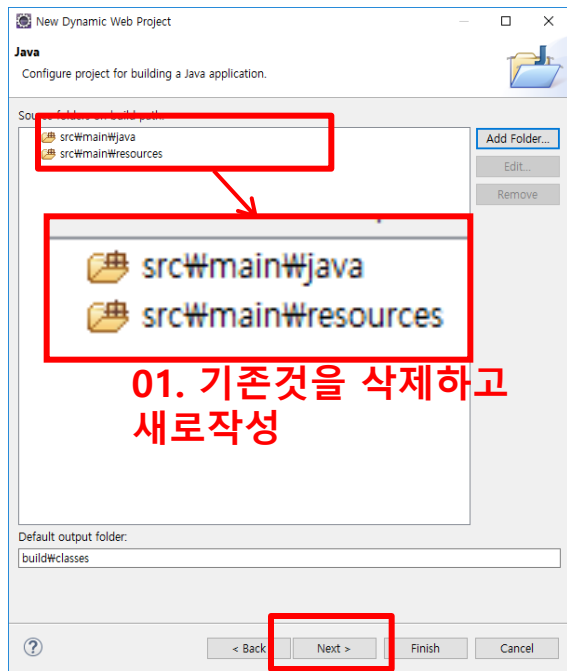
## 02 프로젝트 만들기

### ■ 프로젝트 만들기 - 1

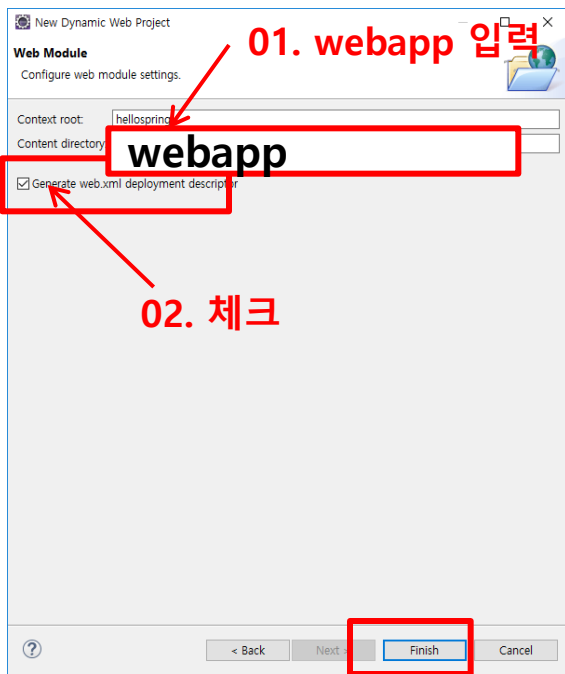


## 02 프로젝트 만들기

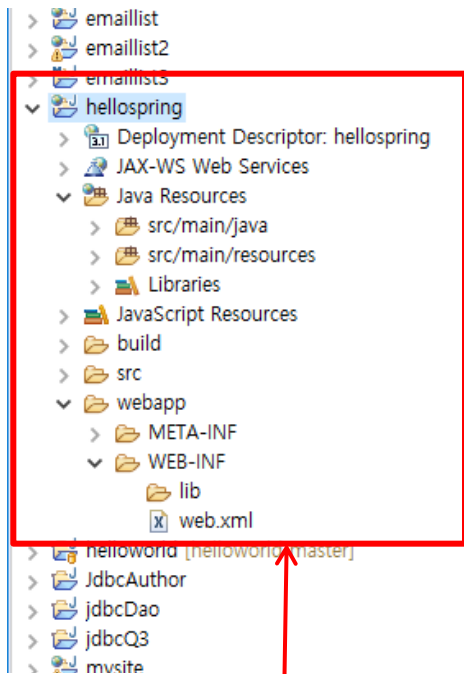
### 프로젝트 만들기 - 2



02. Next 클릭



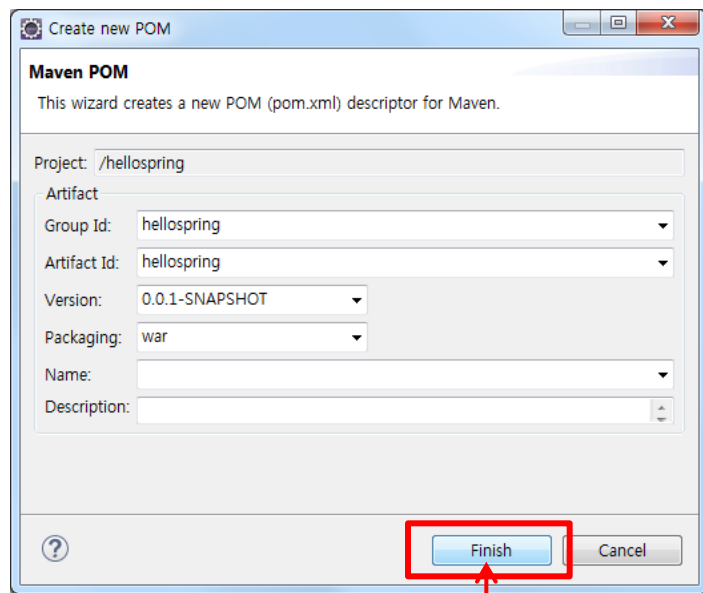
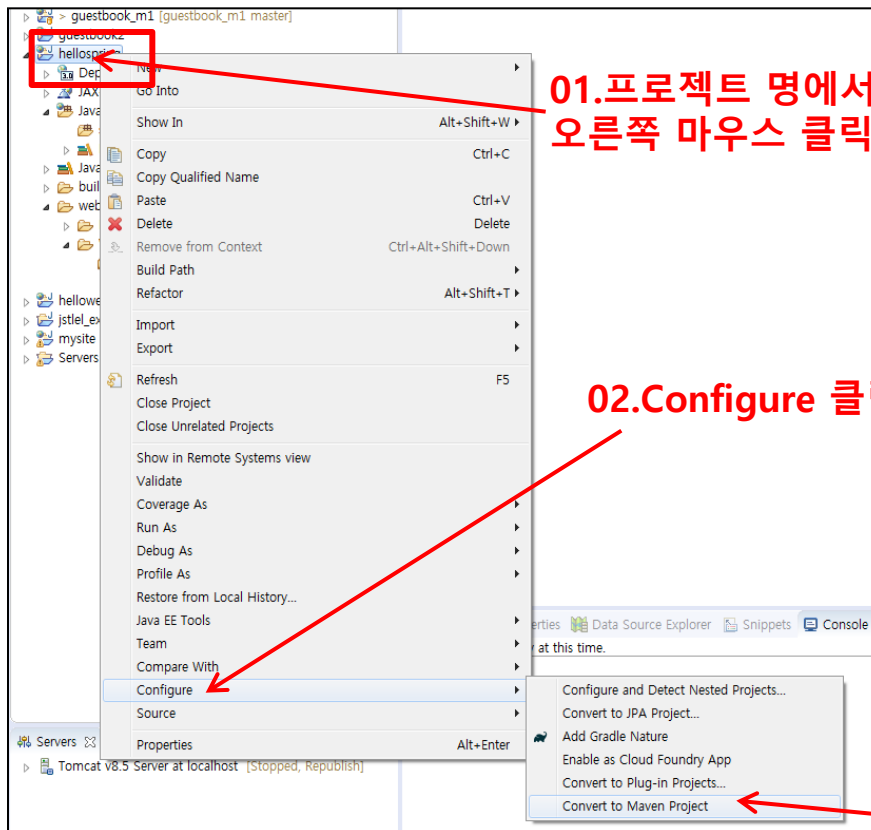
03. Finish 클릭



01. 프로젝트 확인

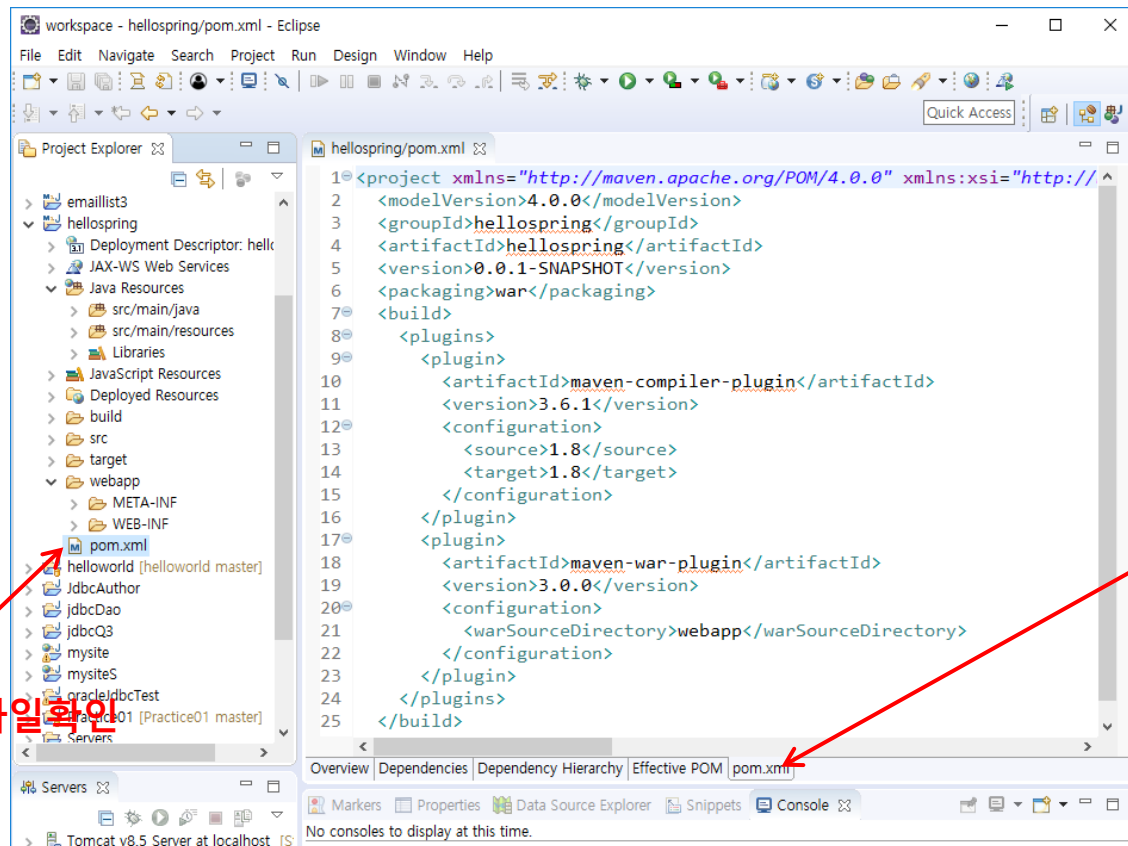
## 02 프로젝트 만들기

### ■ 메이븐 프로젝트로 변경 - 1



## 02 프로젝트 만들기

### ■ 메이븐 프로젝트로 변경 - 2



01.pom.xml 파일확인

02.pom.xml 파일확인

## ■ pom.xml 설정 : Spring Library 의존성 추가

```

1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://maven.apache.org/POM/4.0.0" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
2  <modelVersion>4.0.0</modelVersion>
3  <groupId>hello</groupId>
4  <artifactId>hello</artifactId>
5  <version>0.0.1-SNAPSHOT</version>
6  <packaging>war</packaging>
7
8
9
10
11
12  <build>
13    <sourceDirectory>src</sourceDirectory>
14    <plugins>
15      <plugin>
16        <artifactId>maven-compiler-plugin</artifactId>
17        <version>3.6.1</version>
18        <configuration>
19          <source>1.8</source>

```

붙여넣기

```

<properties>
  <org.springframework-version>4.3.3.RELEASE</org.springframework-version>
</properties>

<dependencies>

  <!-- spring container(core) -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>${org.springframework-version}</version>
  </dependency>

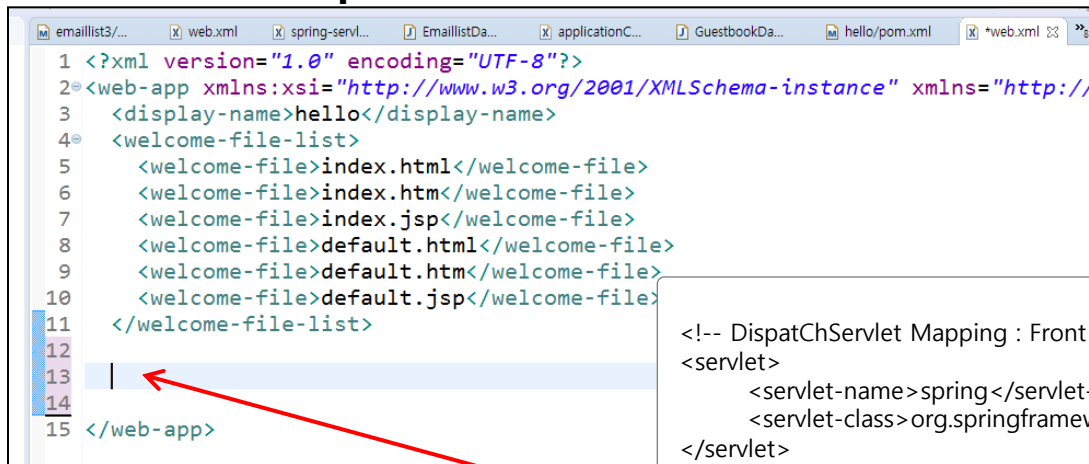
  <!-- Spring Web -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>${org.springframework-version}</version>
  </dependency>

  <!-- Spring MVC -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>${org.springframework-version}</version>
  </dependency>

</dependencies>

```

## ■ web.xml : DispatcherServlet 정의



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://
3   <display-name>hello</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11  </welcome-file-list>
12
13
14
15 </web-app>
```

붙여넣기

<!-- DispatChServlet Mapping : Front Controller 등록 -->

<servlet>

<servlet-name>spring</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>spring</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>



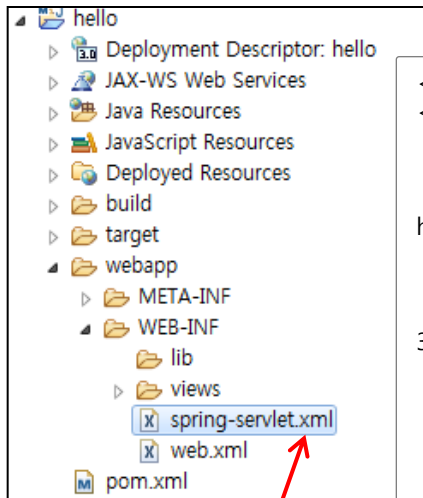
## ■ web.xml : 한글처리

```
<!-- 한글처리 -->
<filter>
  <filter-name>encodingFilter</filter-name>
  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
  <init-param>
    <param-name>encoding</param-name>
    <param-value>UTF-8</param-value>
  </init-param>

  <init-param>
    <param-name>forceEncoding</param-name>
    <param-value>true</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>encodingFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

## ■ /WEB-INF/spring-servlet.xml 만든 후 설정



01.파일 만든 후  
붙여넣기

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:aop="http://www.springframework.org/schema/aop" xmlns="http://www.springframework.org/schema/beans"
       xmlns:p="http://www.springframework.org/schema/p" xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-
3.1.xsd"
       xmlns:mvc="http://www.springframework.org/schema/mvc">

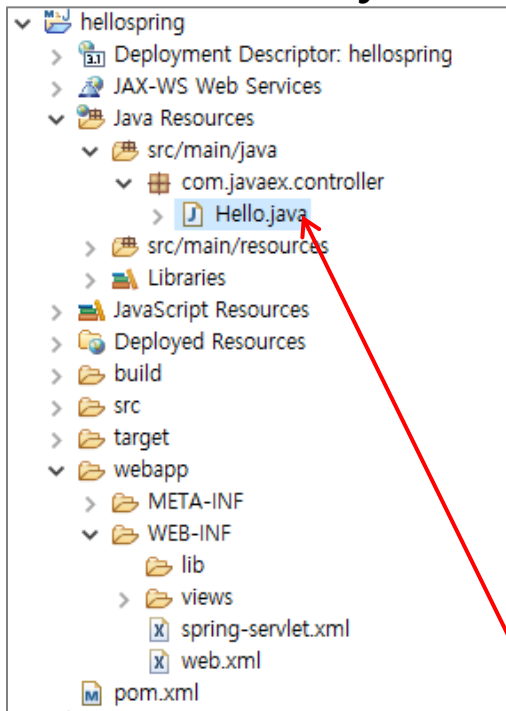
    <!-- annotation 설정을 하겠다. -->
    <context:annotation-config />

    <!--
    com.javaex.controller 패키지 밑에 있는 클래스 중에
    @Controller를 달고 있는 클래스의 객체를 생성 하겠다. (new 하겠다)
    -->

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

</beans>
```

## ■ HelloController.java 파일 만들기 : controller 작성하



```
package com.javaex.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

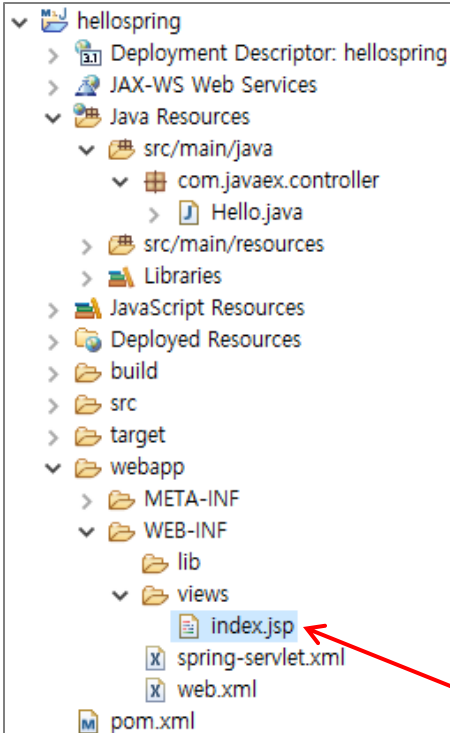
@Controller
public class HelloController{

    @RequestMapping( "/hello")
    public String hello(){
        System.out.println("/hellospring/hello");
        return "/WEB-INF/views/index.jsp";
    }

}
```

**01. com.javaex.controller 패키지 아래에  
Hello.java 만들기**

## ■ index.jsp 파일 만들기 : view 작성하기

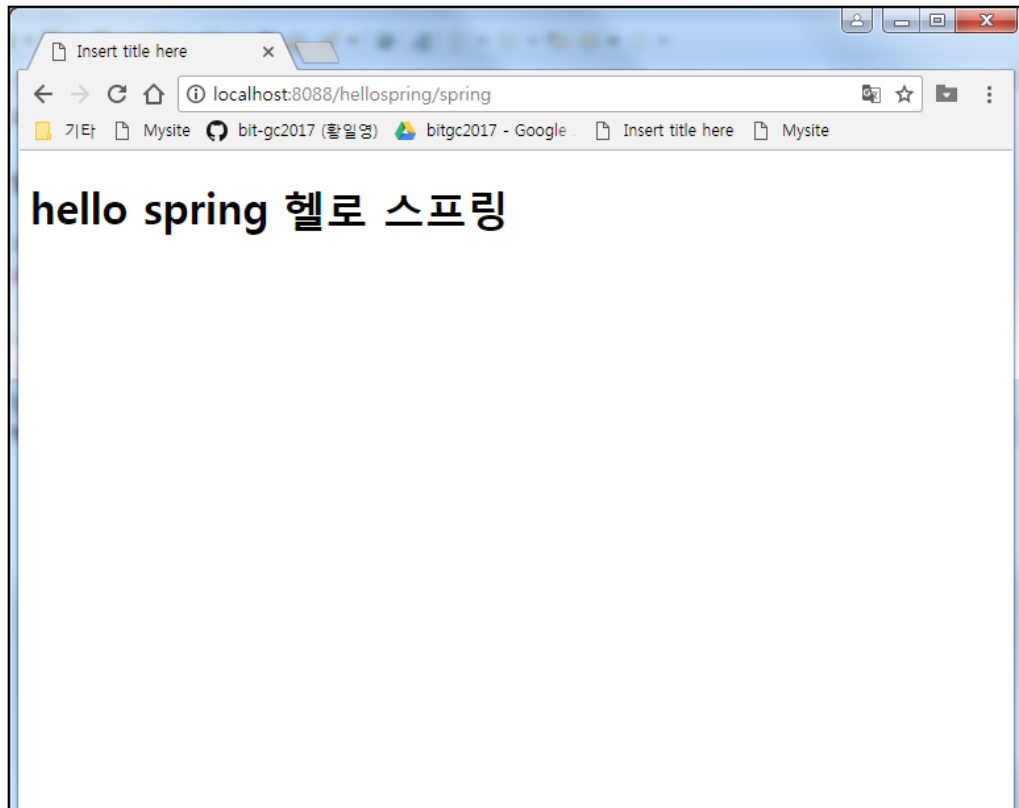


```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
</head>
<body>
    <h1>hello spring    헬로 스프링</h1>
</body>
</html>
```

01./WEB-INF/views/index.jsp 파일 만들기

### ■ 실행화면

# localhost:8088/hellospring/hello



### jstl 라이브러리 추가

## ■ pom.xml 설정 : jstl 라이브러리 추가

```
<!-- jstl -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

<https://mvnrepository.com/>

### jdbc 라이브러리 추가



### ■ pom.xml 설정 : jdbc 라이브러리 추가

```
<!-- Oracle JDBC -->
<dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.4.0-atlassian-hosted</version>
</dependency>
```

```
<repositories>
    <repository>
        <id>oracle</id>
        <name>ORACLE JDBC Repository</name>
        <url>https://maven.atlassian.com/3rdparty/</url>
    </repository>
</repositories>
```

```
<!-- Oracle JDBC https://repo1.maven.org/maven2/ -->
<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc6 -->
<dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.4</version>
</dependency>
```

잘 안되는 경우

<https://mvnrepository.com/> 참고해서 적절히 추가 후 Maven Project Update 해본다

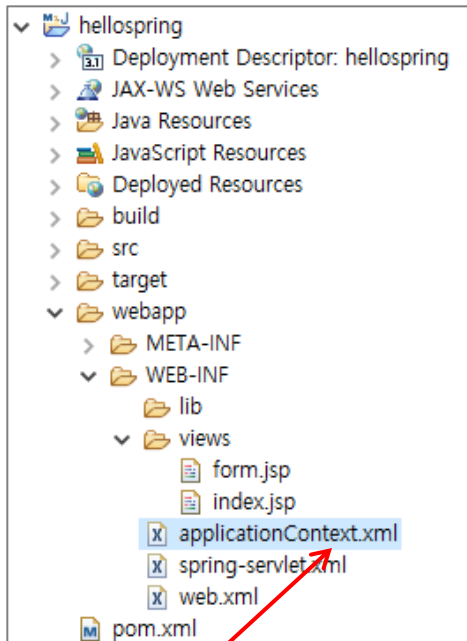
## applicationContext 설정

## ■ web.xml : Context Listener 등록

```
<!-- Context Listener 등록 -->
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext.xml</param-value>
</context-param>
```

## ■ applicationContext 파일 만들기



01.applicationContext.xml

```
<?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:aop="http://www.springframework.org/schema/aop"
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/aop http://www.springframework.org/schema/aop/spring-aop.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee.xsd
http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd
http://www.springframework.org/schema/data/mongo http://www.springframework.org/schema/data/mongo/spring-mongo-1.1.xsd
http://ehcache-spring-annotations.googlecode.com/svn/schema/ehcache-spring
http://ehcache-spring-annotations.googlecode.com/svn/schema/ehcache-spring/ehcache-spring-1.1.xsd
http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-3.0.xsd">
```

<context:annotation-config />

<context:component-scan base-package="com.emailist">

<context:include-filter type="annotation" expression="org.springframework.stereotype.Repository" />

<context:include-filter type="annotation" expression="org.springframework.stereotype.Service" />

<context:include-filter type="annotation" expression="org.springframework.stereotype.Component" />

</context:component-scan>

</beans>

## html, css, js, img 파일 접근 처리

## ■ html, css, js, img 파일 접근시 오류

- DefaultServlet이 핸들러를 찾을 수 없기 때문

## ■ 매핑정보 중에서 스프링이 처리할 수 없는 요청은 서블릿 컨테이너(톰캣)이 제공하는 디폴트 서블릿에게 위임 시키기. ( servlet.xml )

### spring-servlet.xml

```
<!-- 서블릿 컨테이너의 디폴트 서블릿 위임 핸들러 -->
<mvc:default-servlet-handler />
<mvc:annotation-driven />
```

## ViewResolver 설정

### spring-servlet.xml

```
<!-- 기본 뷰 리졸버 -->
<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="viewClass" value="org.springframework.web.servlet.view.JstlView" />
    <property name="prefix" value="/WEB-INF/views/" />
    <property name="suffix" value=".jsp" />
    <property name="order" value="1" />
</bean>
```

# chapter01

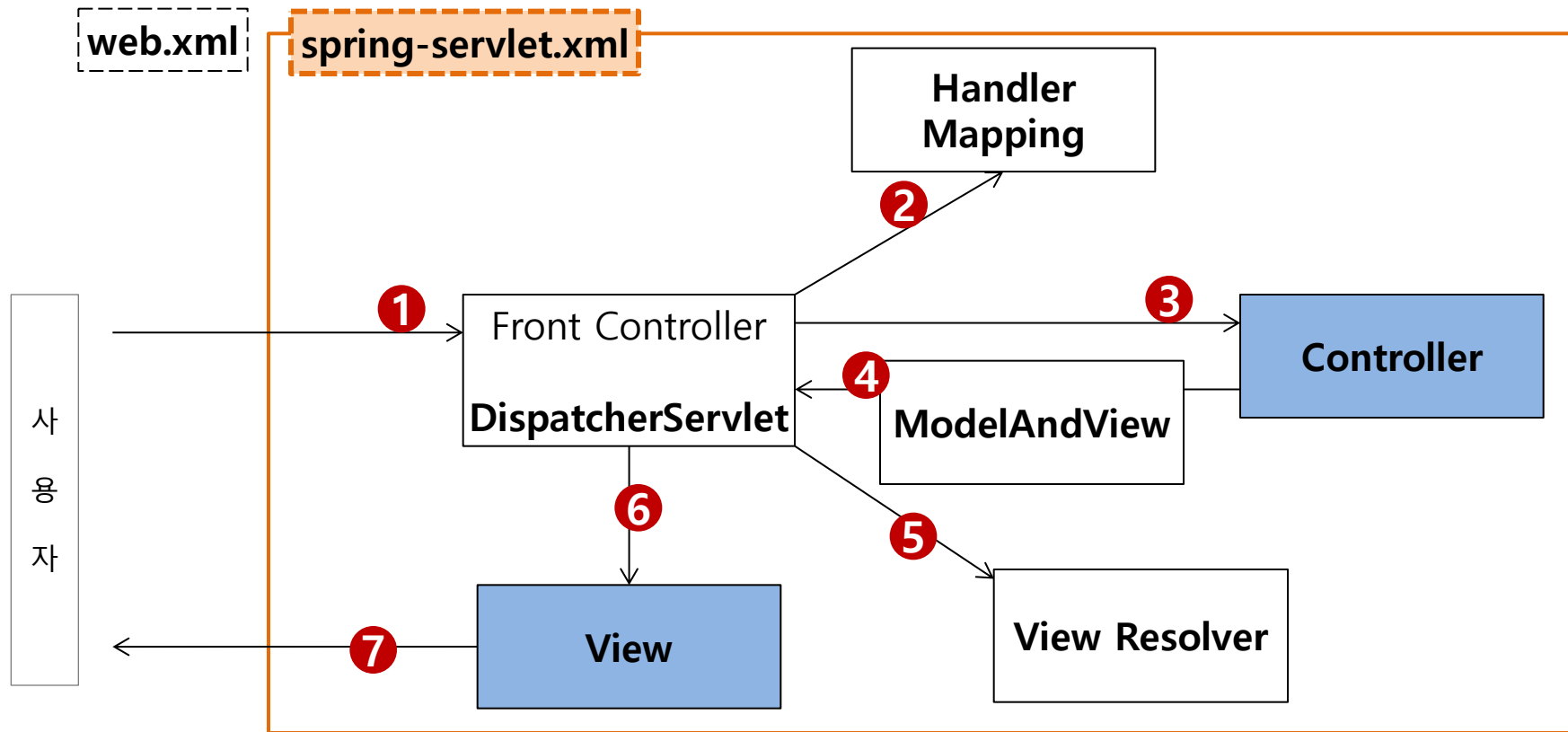
## spring 시작하기

1. spring 개요
2. 프로젝트 만들기 (hello spring)
3. 스프링의 **DispatcherServlet** 과 **MVC**



## 03 DispatchedServlet 과 MVC

### ■ 스프링의 DispatchedServlet 과 MVC



## ■ 스프링의 DispatchedServlet 과 MVC

- 사용자의 요청을 DispatcherServlet이 받는다.
- 요청을 처리해야 하는 컨트롤을 찾기위해 HandlerMapping에게 질의를 하고 HandlerMapping은 컨트롤 객체에 매핑되어 있는 URL를 찾아낸다.
- Dispatcher Servlet은 찾은 컨트롤에게 요청을 전달하고 Controller는 서비스 계층의 인터페이스를 호출하여 적절한 비즈니스를 수행한다.
- 컨트롤러는 비즈니스 로직의 수행결과로 받아낸 도메인 모델 객체와 함께 뷰이름을 ModelAndView 객체에 저장하여 리턴한다.
- DispatcherServlet은 응답할 View를 찾기위해 ViewResolver에게 질의를 한다.
- DispatcherServlet은 찾아낸 View 객체에게 요청을 전달한다.

## 03 DispatchedServlet 과 MVC

### ■ 추가적으로 해준것들

- pom.xml 구성
- DispatcherServlet 등록( web.xml )
- 서블릿 애플리케이션 컨텍스트 설정 (spring-servlet.xml)
- Controller 작성

### ■ 생략된 것들

- 서블릿 작성
- forwarding
- ...