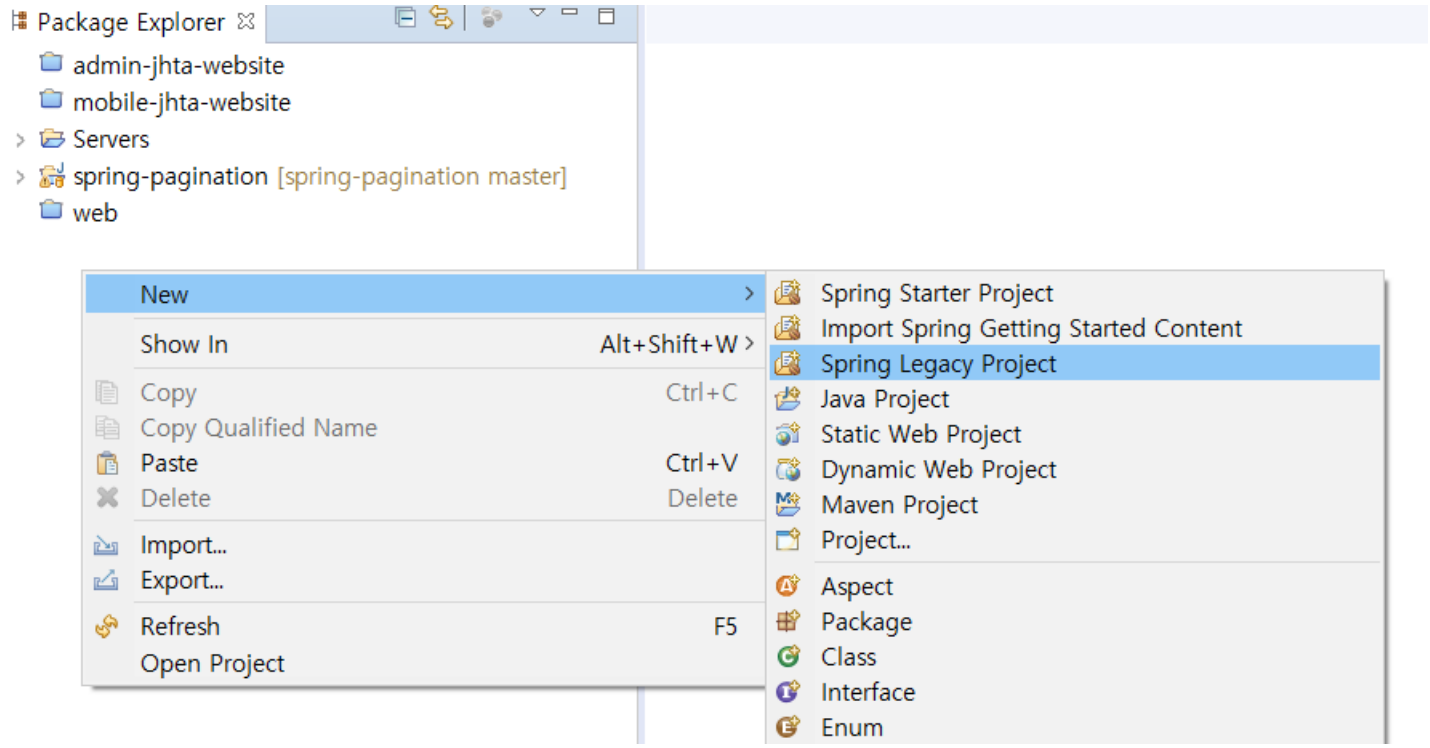
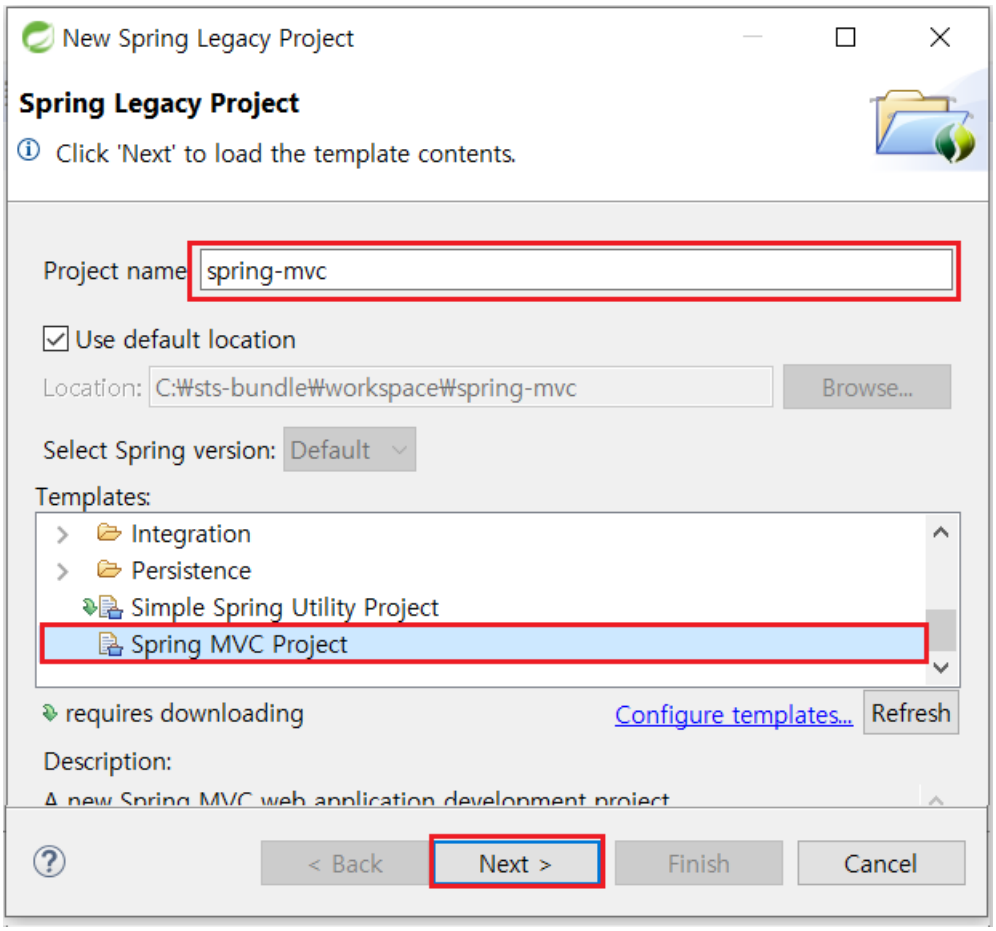
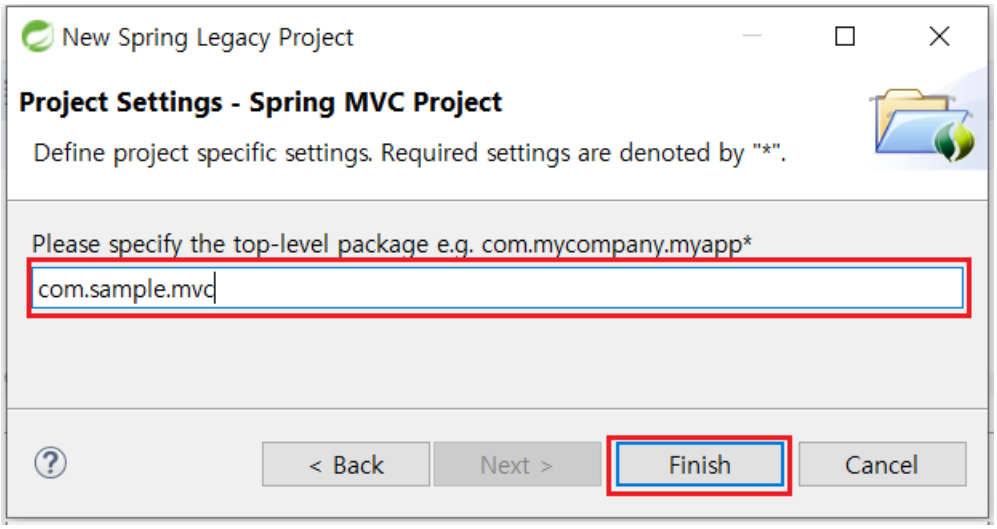
프로젝트 생성하기



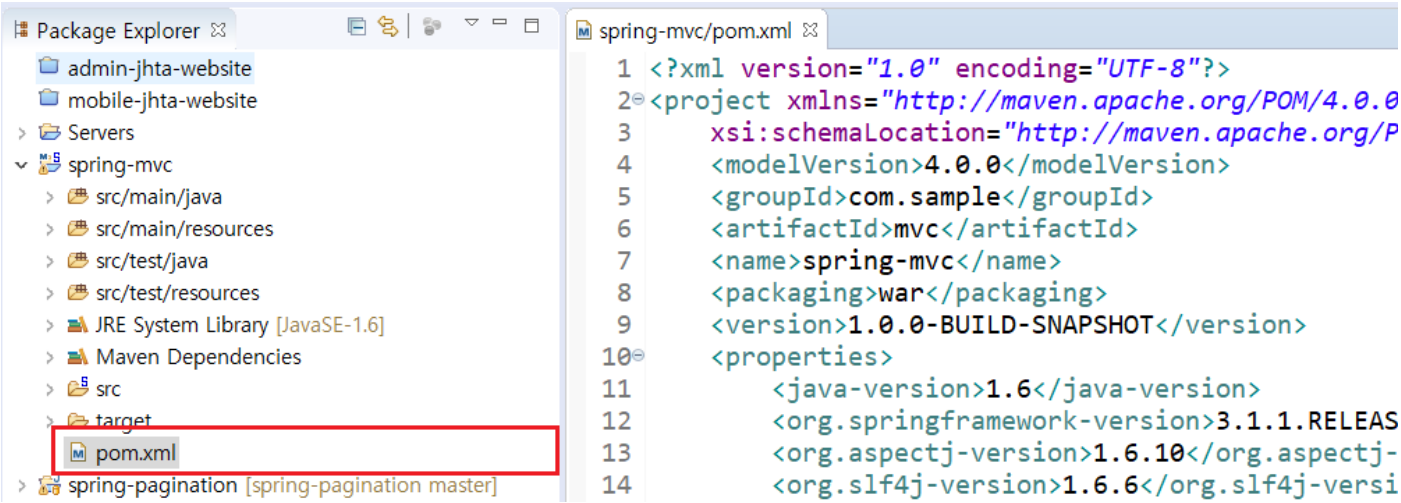

Spring MVC 프로젝트 선택하기




프로젝트의 기본 패키지 정의




프로젝트 설정파일 변경하기




프로젝트 설정파일의 <properties> 설정하기

```
<properties>
    <!--  자바 소스 및 컴파일러 버전 설정하기 -->
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <!--  프로젝트 소스 및 출력 인코딩 방식 설정하기 -->
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <!--  스프링프레임워크 버전 설정하기 -->
    <springframework-version>4.3.24.RELEASE</springframework-version>
    <!--  Aspectj 버전 설정하기 -->
    <aspectj-version>1.9.4</aspectj-version>
</properties>
```

프로젝트 설정파일의 <repositories> 설정하기

```xml
<repositories>
    <!-- 스프링 JDBC 드라이버를 제공하는 메이븐 원격 저장소 설정하기 -->
    <repository>
        <id>codelds</id>
        <url>https://code.lds.org/nexus/content/groups/main-repo</url>
    </repository>
</repositories>
```

프로젝트 설정파일의 <dependencies>에 스프링프레임워크 관련 의존성 설정하기

```xml
<dependencies>

    <!-- 프로젝트를 위한 스프링프레임워크 의존성 추가하기 -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>${springframework-version}</version>
    </dependency>
    <!-- 프로젝트를 위한 스프링프레임워크 데이터베이스 엑세스 관련 의존성 추가하기-->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>${springframework-version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-tx</artifactId>
        <version>${springframework-version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-orm</artifactId>
        <version>${springframework-version}</version>
    </dependency>
    <!-- 프로젝트를 위한 스프링프레임워크 MVC 의존성 추가하기 -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>${springframework-version}</version>
    </dependency>

</dependencies>
```

프로젝트 설정파일의 <dependencies>에 데이터베이스 엑세스 관련 의존성 설정하기

```xml
<!-- 프로젝트를 위한 Oracle JDBC 드라이브 의존성 추가하기 -->
<dependency>
    <groupId>com.oracle</groupId>
    <artifactId>ojdbc6</artifactId>
    <version>11.2.0.3</version>
</dependency>
<!-- 프로젝트를 위한 Connection Pool 관련 의존성 추가하기 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.5.0</version>
</dependency>
<!-- 프로젝트를 위한 mybatis오 mybatis-spring 연동 관련 의존성 추가하기 -->
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.2</version>
</dependency>
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.2</version>
</dependency>
```

프로젝트 설정파일의 <dependencies>에 기타 의존성 설정하기

```xml
<!-- 프로젝트를 위한 선언적트랜잭션처리에 필요한 AOP관련 의존성 추가하기 -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>${aspectj-version}</version>
</dependency>
<!-- 프로젝트를 위한 파일업로드 지원 의존성 추가하기 -->
<dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
    <version>1.3.3</version>
</dependency>
<!-- 프로젝트를 위한 자바객체와 JSON 텍스트 사이의 변환의 지원하는 의존성 추가하기 -->
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.9.9.1</version>
</dependency>
```

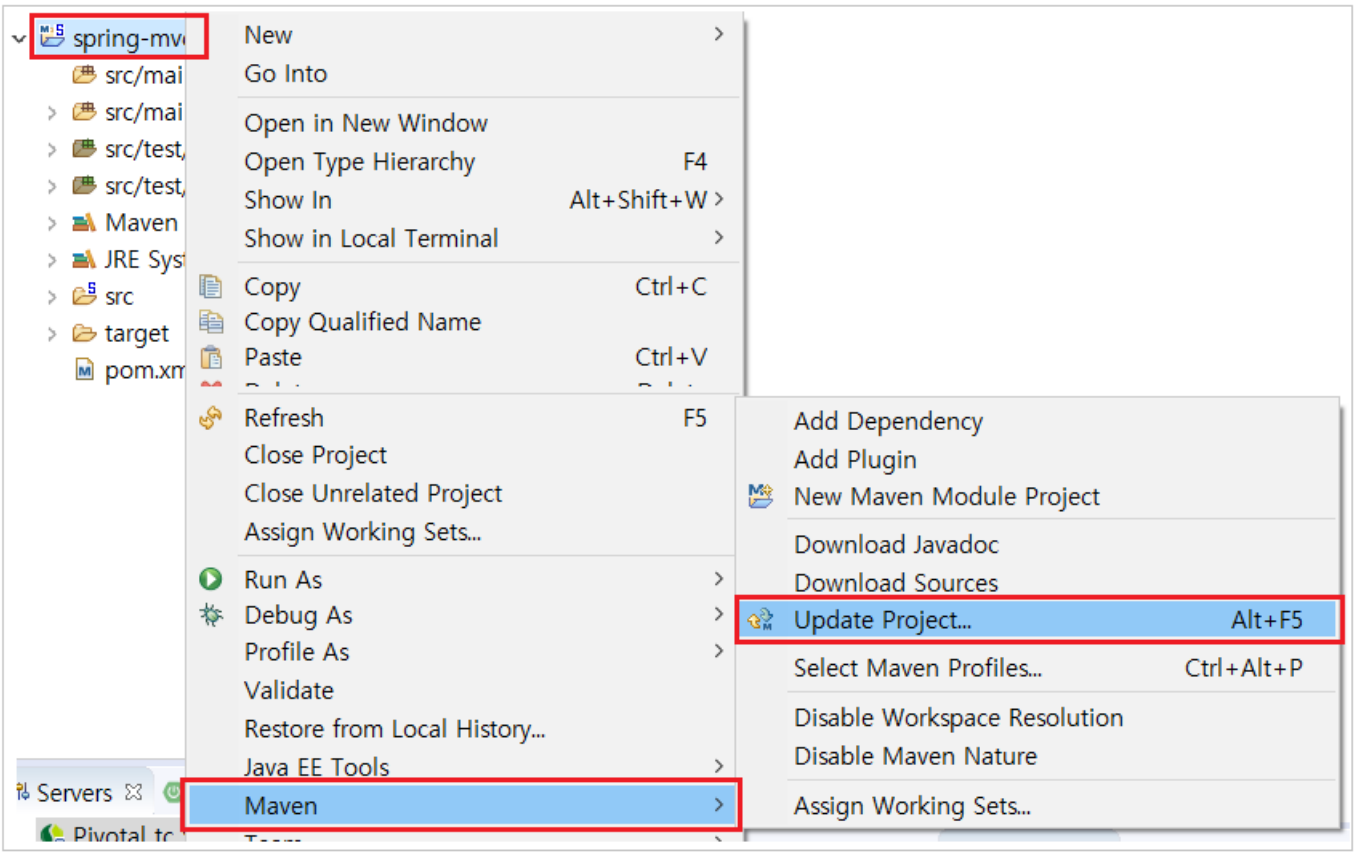
프로젝트 설정파일의 <dependencies>에 로깅관련 의존성 설정하기

```xml
<!-- 프로젝트를 위한 로그출력을 지원하는 의존성 추가하기 -->
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-api</artifactId>
    <version>1.7.28</version>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>jcl-over-slf4j</artifactId>
    <version>1.7.28</version>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>1.7.28</version>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.apache.logging.log4j</groupId>
    <artifactId>log4j-core</artifactId>
    <version>2.12.1</version>
</dependency>
```

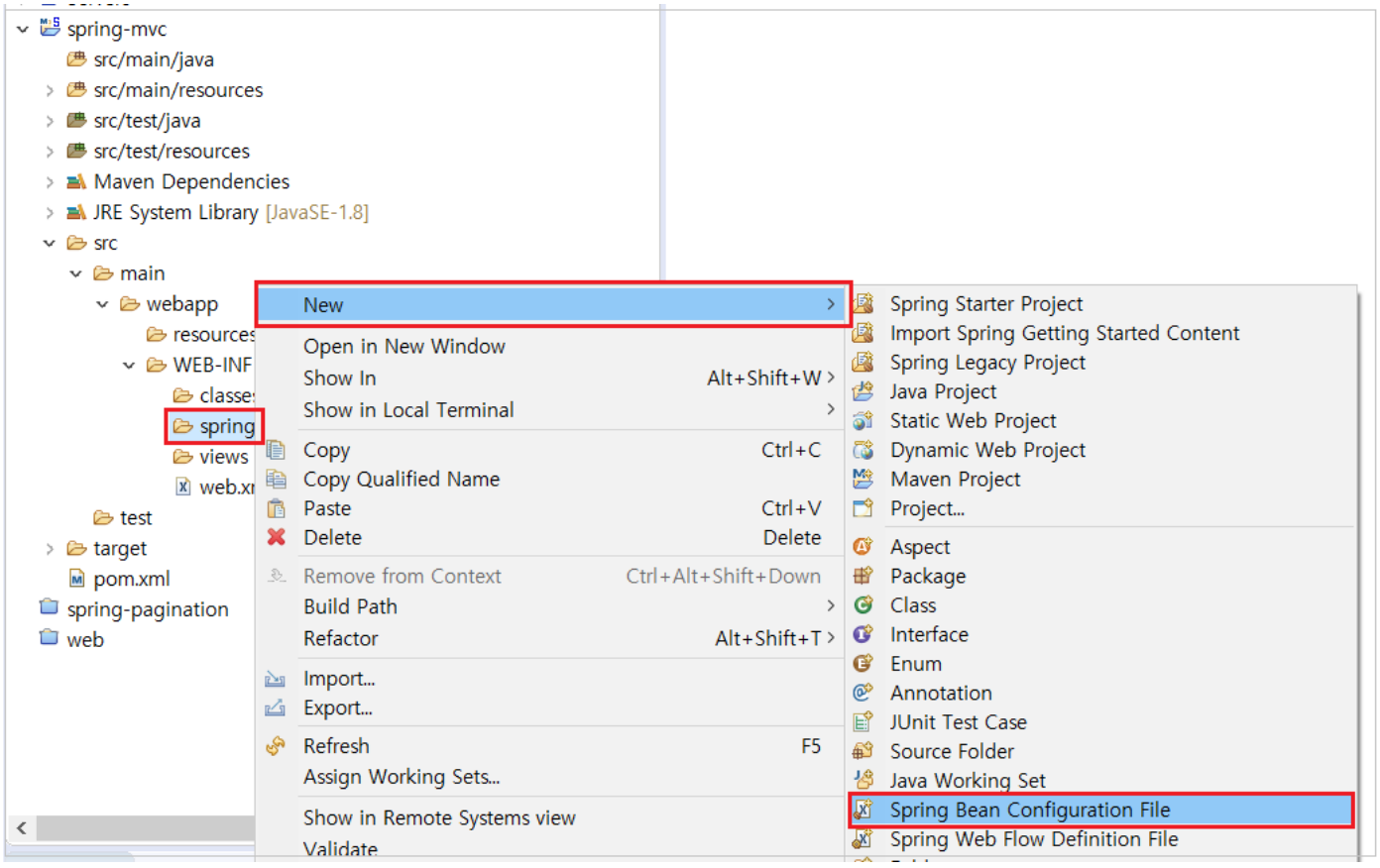
프로젝트 설정파일의 <dependencies>에 웹 관련(서블릿, JSP, JSTL) 의존성 설정하기

```xml
<!-- 프로젝트를 위한 서블릿/JSP/JSTL 의존성 추가하기 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>2.5</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.1</version>
    <scope>provided</scope>
</dependency>
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

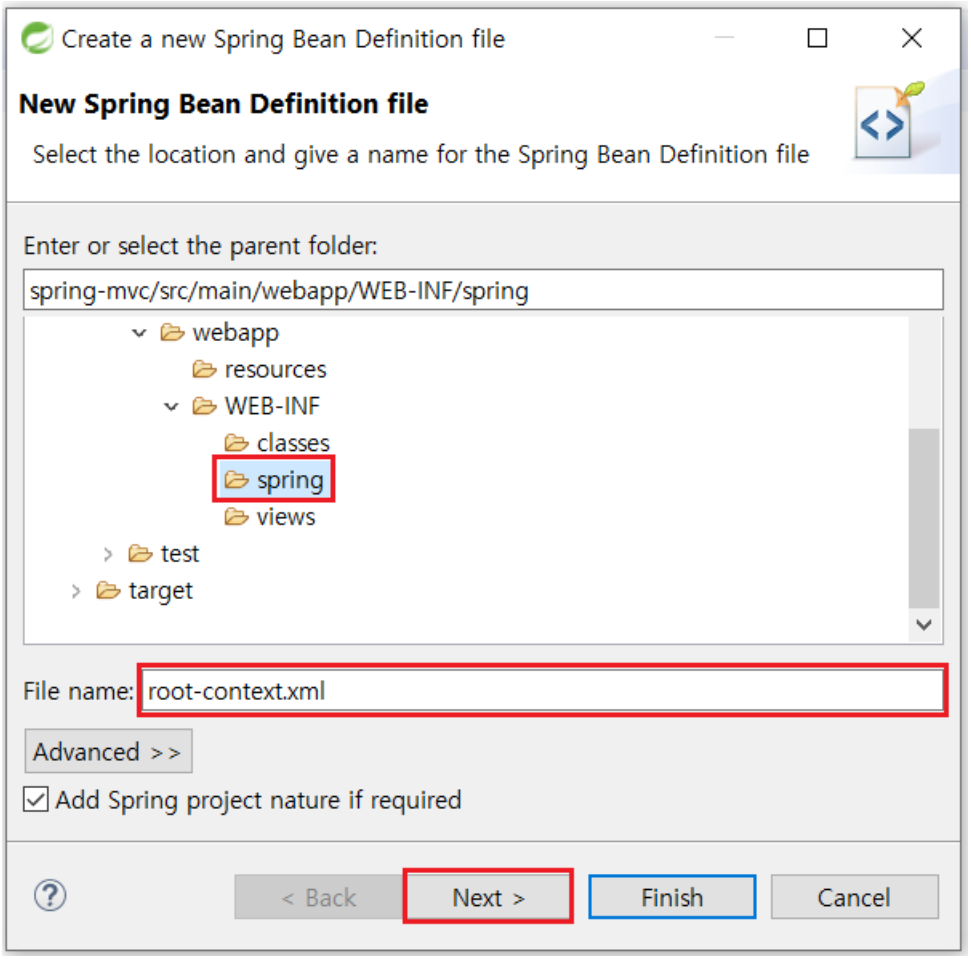
프로젝트 설정파일의 정보를 프로젝트에 반영하기



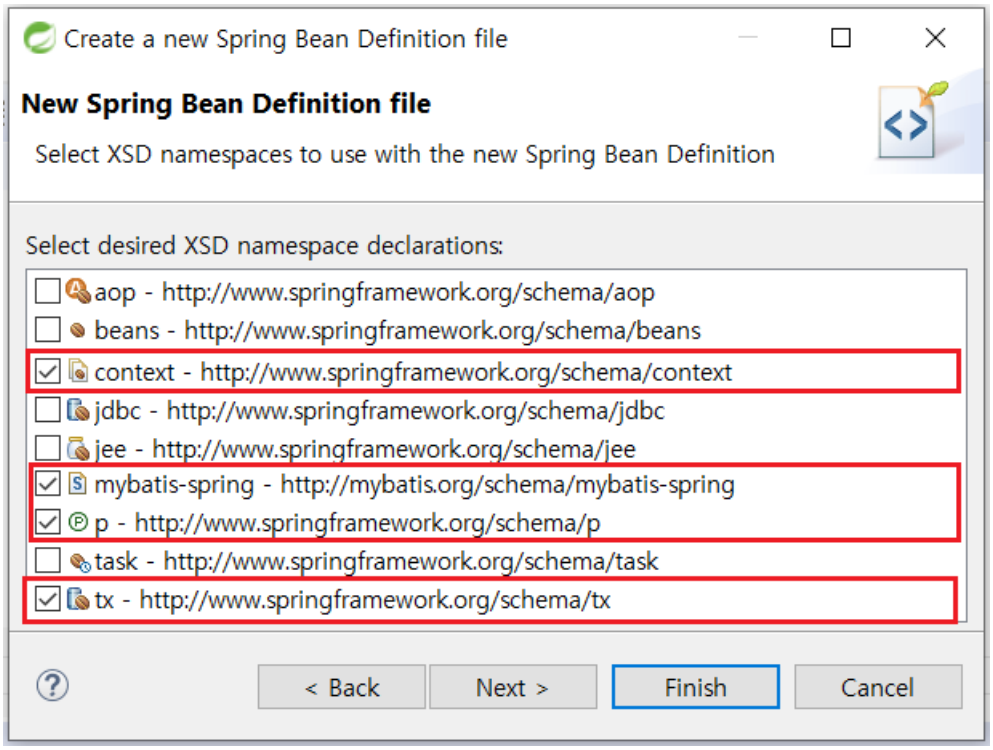

스프링 빈 설정파일(Root 스프링 컨테이너) 생성하기



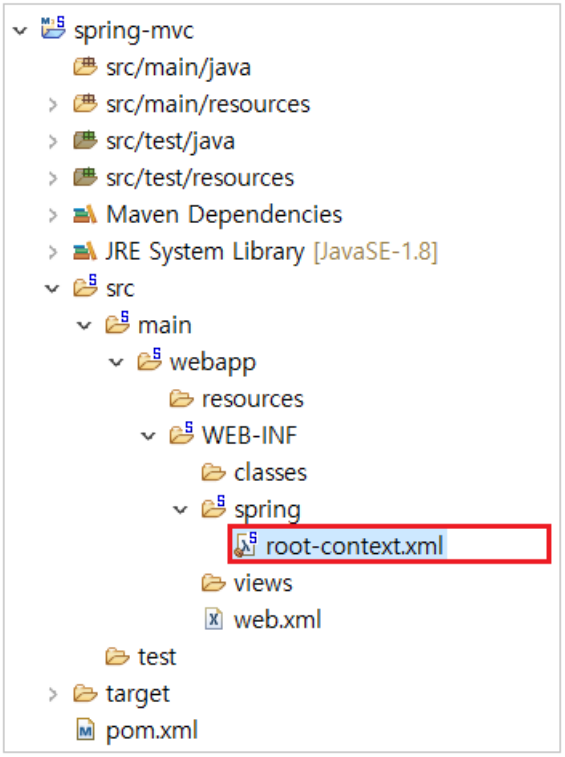

스프링 빈 설정파일 이름 지정하기



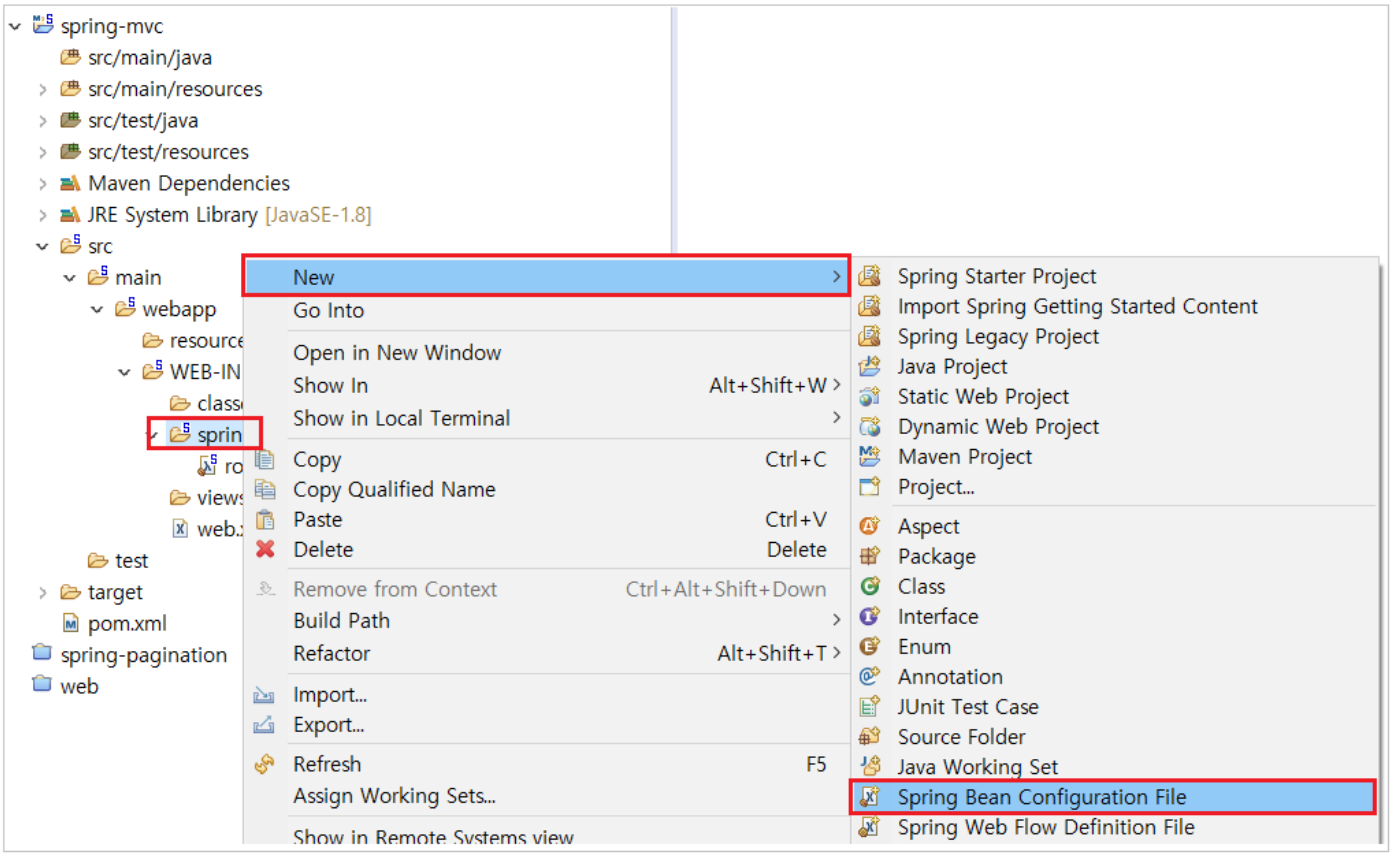

스프링 빈 설정파일(Root 스프링 컨테이너) 네임스페이스 지정하기



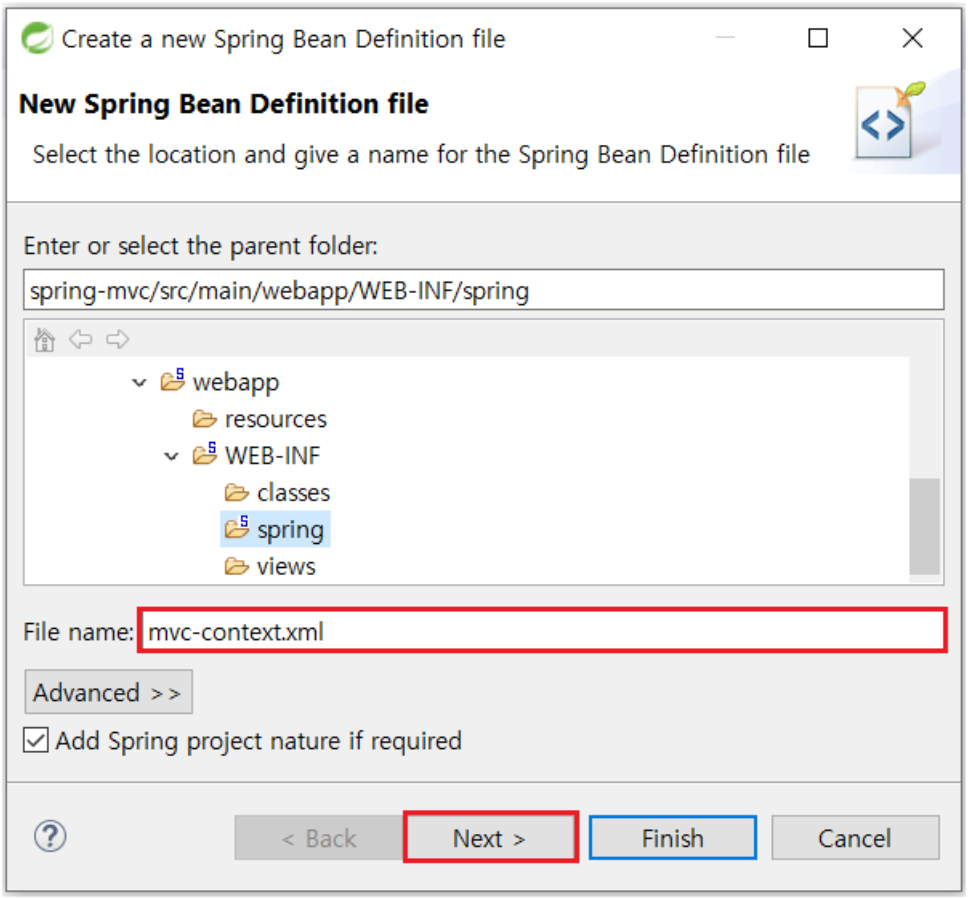

스프링 빈 설정파일(Root 스프링 컨테이너) 생성확인



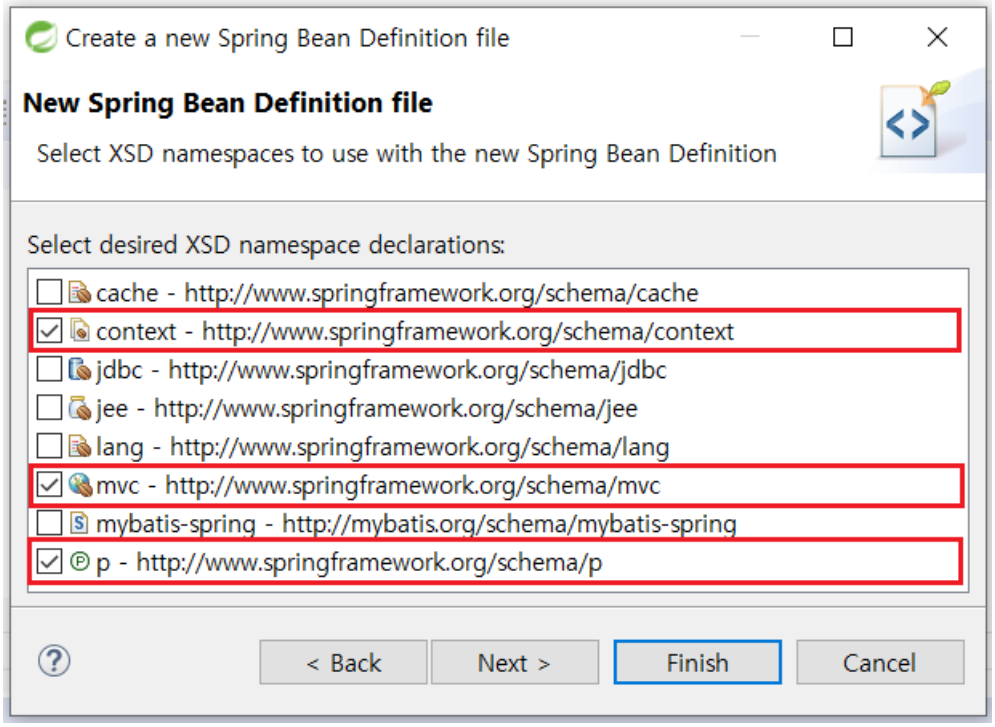

스프링 빈 설정파일(Child 스프링 컨테이너) 생성하기



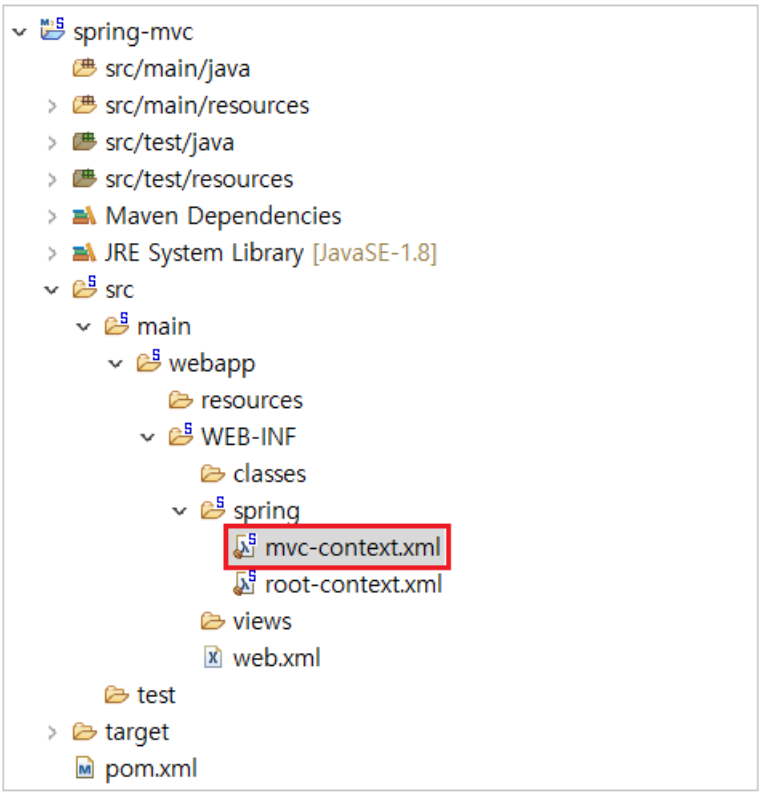

스프링 빈 설정파일(Child 스프링 컨테이너) 이름 정하기




스프링 빈 설정파일(Child 스프링 컨테이너)의 네임스페이스 지정하기




스프링 빈 설정파일(Child 스프링 컨테이너) 생성 확인하기




프로젝트의 web.xml에 스프링 관련 설정정보 추가하기

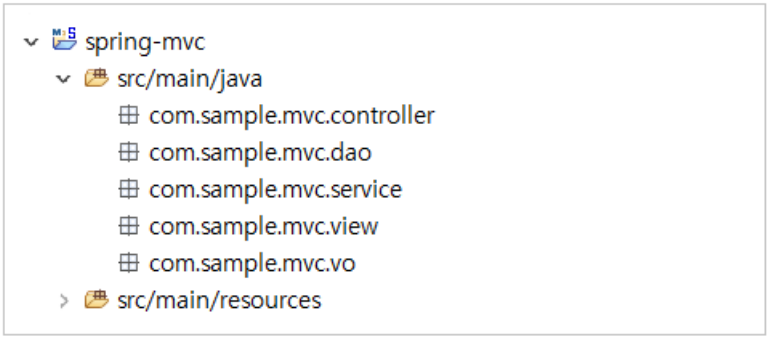
```
<!-- Root Spring Container 설정하기 -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml</param-value>
</context-param>
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

```
<!-- Child Spring Container 설정하기 -->
<servlet>
    <servlet-name>app</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring/mvc-context.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>app</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

프로젝트의 web.xml에 필터정보 추가하기

```xml
<!-- 요청 파라미터 한글처리를 위한 필터 등록 -->
<filter>
    <filter-name>characterEncodingFilter</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>
</filter>
<filter-mapping>
    <filter-name>characterEncodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

패키지 생성하기




HomeController 생성하기

```java
package com.sample.mvc.controller;

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

@Controller
public class HomeController {

    @GetMapping("/")
    public String home() {

        return "home";
    }
}
```

/webapp/resources에 부트스트랩과 jQuery 라이브러리 추가하기

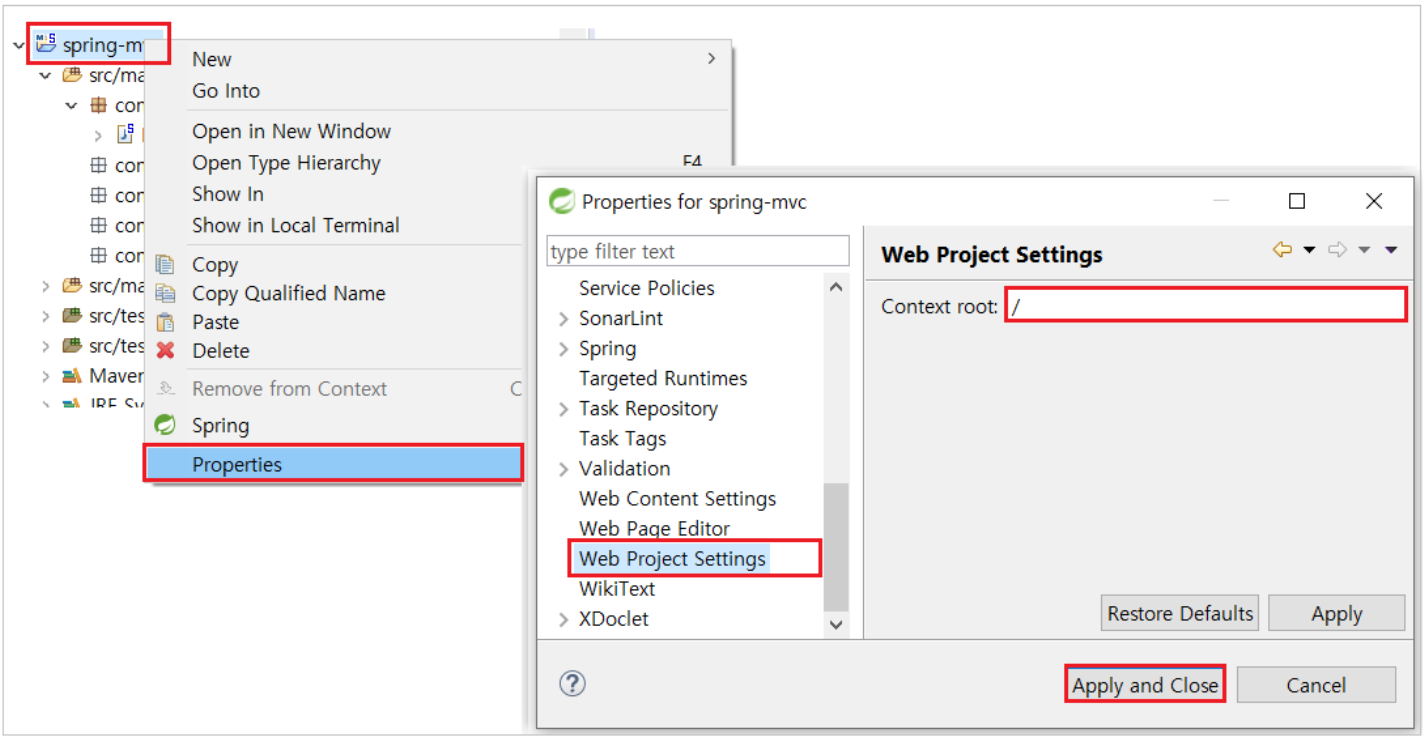

/WEB-INF/views/home.jsp 생성하기

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="ko">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="/resources/bootstrap/css/bootstrap.min.css">
    <script src="/resources/jquery/jquery.min.js"></script>
    <script src="/resources/bootstrap/js/bootstrap.min.js"></script>
</head>
<body>

<div class="container">
    <div class="page-header">
        <h1>Home Page</h1>
    </div>
</div>

</body>
</html>
```

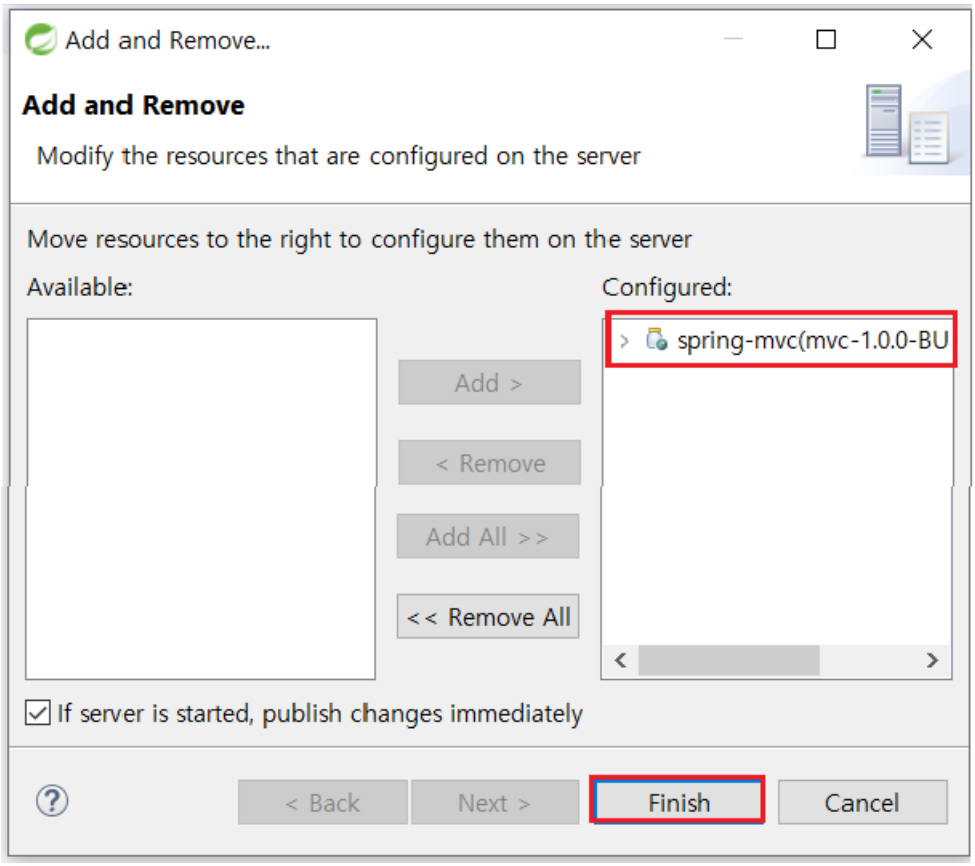
/WEB-INF/spring/mvc-context.xml에 SpringMVC 설정 추가하기

```
<!-- Spring MVC 관련 @어노테이션 설정 정보를 분석하고 적절한 처리를 수행하는 빈을 등록한다-->
<mvc:annotation-driven />
<!-- 정적리소스(css, js, image)는 /resources/**으로 요청하면 /resources/의 리소스를 응답으로 제공한다-->
<mvc:resources mapping="/resources/**" location="/resources/" />

<!-- JSP를 실행해서 동적 컨텐츠(HTML)을 응답으로 제공하는 InternalResourceView를 제공하는 뷰리졸브를 등록한다. -->
<mvc:view-resolvers>
    <!-- 실행할 JSP는 컨트롤러가 반환하는 뷰이름에 /WEB-INF/views/와 .jsp를 붙여서 완성한다. -->
    <mvc:jsp prefix="/WEB-INF/views/" suffix=".jsp"/>
</mvc:view-resolvers>

<!-- com.sample.mvc.controller 패키지에서 클래스를 스캔해서 스프링의 빈으로 등록한다. -->
<context:component-scan base-package="com.sample.mvc.controller" />
```

프로젝트의 Context Root를 변경하기




프로젝트를 톰캣에 배포하고 서버를 실행한다.




브라우저에서 시작화면 요청하기