



박 경 태

comsi.java@gmail.com

고급 자바 프로그래밍
: STS를 이용한 Spring 프로그래밍

강의 내용

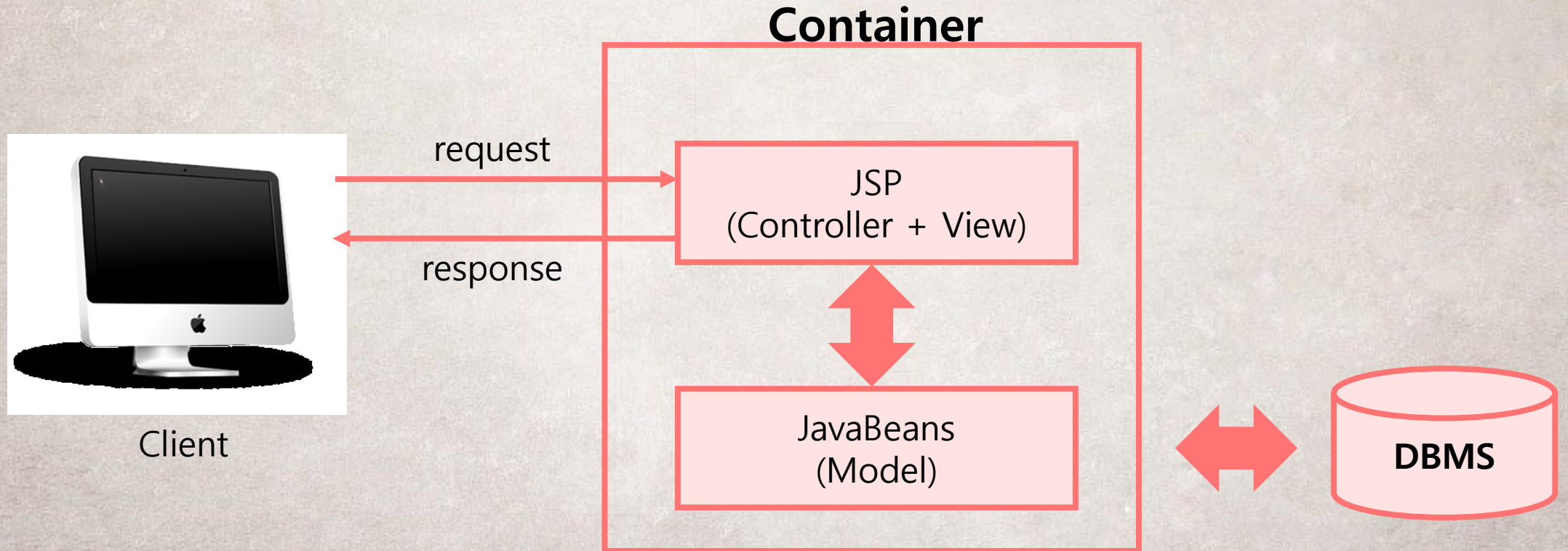
순서	내 용
1	<ul style="list-style-type: none">• Spring IoC를 이용한 비즈니스 컴포넌트 만들기
2	<ul style="list-style-type: none">• Spring AOP(Aspect Oriented Programming)를 이용한 공통 서비스 만들기• Spring DAO(Data Access Object)를 이용한 데이터베이스 연동 및 트랜잭션 처리
3	<ul style="list-style-type: none">• Spring MVC를 이용한 MVC 아키텍처 적용하기
4	<ul style="list-style-type: none">• Spring MVC의 부가 기능 사용하기(파일 업로드, 다국어, 예외 처리 등)
5	<ul style="list-style-type: none">• Spring과 MyBatis 연동하기• Spring과 JPA 연동하기

Model 1 아키텍처로 게시판 개발

Model 1 아키텍처 구조

- Model 1 아키텍처는 JSP와 JavaBean만 사용하여 웹 개발

Model 1 아키텍처 구조



➔ 자바에서 Bean은 객체를 의미함.

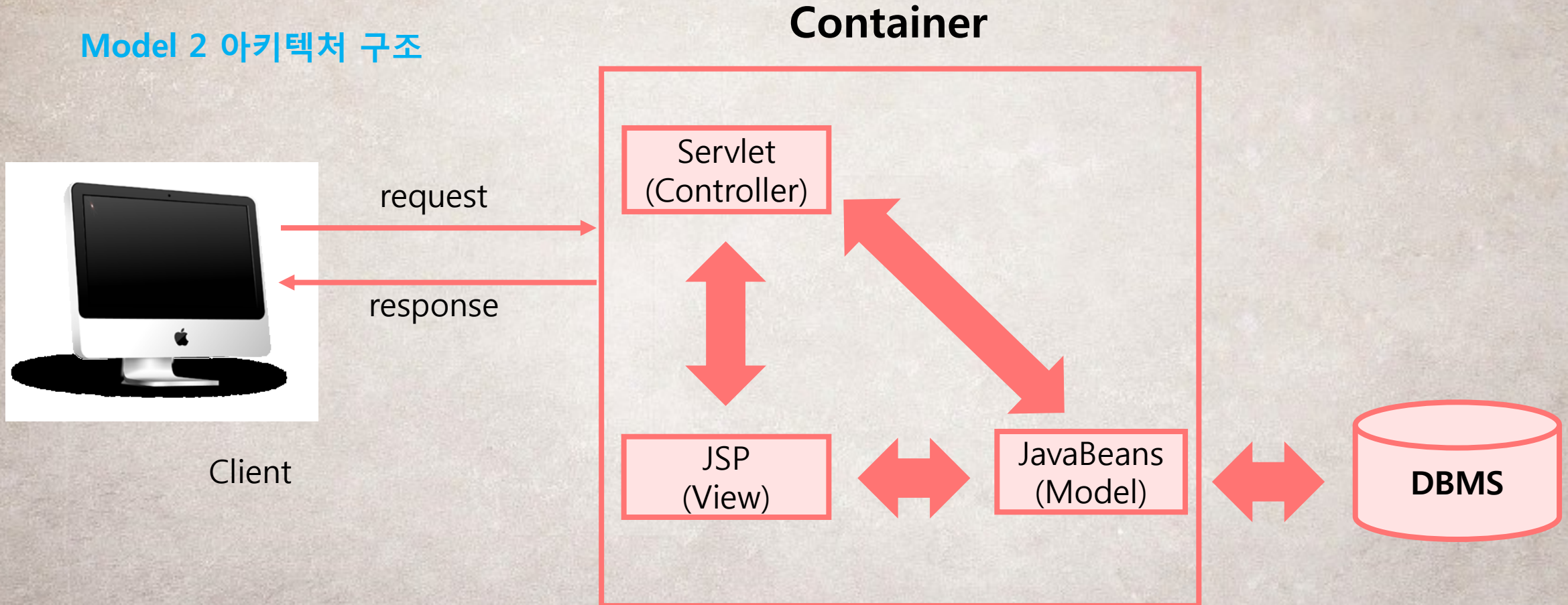
Model 1 아키텍처 구조

- **Model** : 데이터베이스 연동 로직을 제공하면서 DB에서 검색한 데이터가 저장되는 자바 객체
 - 스프링 IoC와 AOP 실습을 하면서 VO와 DAO 클래스를 사용했으며, 이 두 클래스가 바로 Model 기능의 자바 객체
 - 모델1에서는 JSP 파일이 가장 중요한 역할을 수행 - Controller와 View
 - Controller는 사용자의 요청 처리와 관련된 자바 코드
 - View는 사용자와 상호작용하는 영역
 - JSP는 Model을 사용하여 검색한 데이터를 사용자가 원하는 화면으로 제공하기 위해 다양한 마크업(markup)을 사용한다. 마크업 언어로는 HTML, CSS가 사용된다.
 - 결과적으로 JSP 파일에 자바 코드와 마크업 관련 코드들이 섞임
 - 역할구분이 불명확해지고 JSP 파일에 대한 디버깅과 유지보수가 어려워짐
- ➔ 적은 개발 인력으로 간단한 프로젝트에 사용 가능하지만 엔터프라이즈급의 복잡한 시스템에는 부적절한 모델

Model 2 아키텍처(MVC) 등장

- Model 1 구조의 단점을 보완하기 위해 만들어진 구조

Model 2 아키텍처 구조



Model 2 아키텍처 구조

- Model 1은 엔터프라이즈 시스템에 적합하지 않은 이유로 자바 로직과 화면 디자인이 통합되어 유지보수가 어렵다는 것.
- Model 1의 문제를 해결하기 위해 고안된 웹 개발 모델이 Model2(MVC)
- Model 2는 가장 큰 특징은 Controller의 등장이며 이는 서블릿 클래스를 중심으로 구현됨
 - Model 1의 JSP에서 자바 코드만 Controller로 이동하면 Model 2가 됨
- MVC 아키텍처에서 각요소의 기능과 개발 주체

기능	구성요소	개발주체
Model	VO, DAO 클래스	자바 개발자
View	JSP 페이지	웹디자이너
Controller	Servlet	자바 개발자 또는 MVC 프레임워크

- MVC아키텍처에서 가장 중요한 부분이 Controller인데, 이 Controller를 성능과 유지보수의 편의성을 고려하여 잘 만드는 것이 중요함

모델2를 이용한 게시판 구현

- JSP를 이용한 View 페이지

Project folder structure

- **src/main/java** : This folder has java sources.
- **src/main/resources**: This folder can hold property files and other resources
- **src/main/webapp**: This folder holds jsp and other web application content. ➔ 추가필요

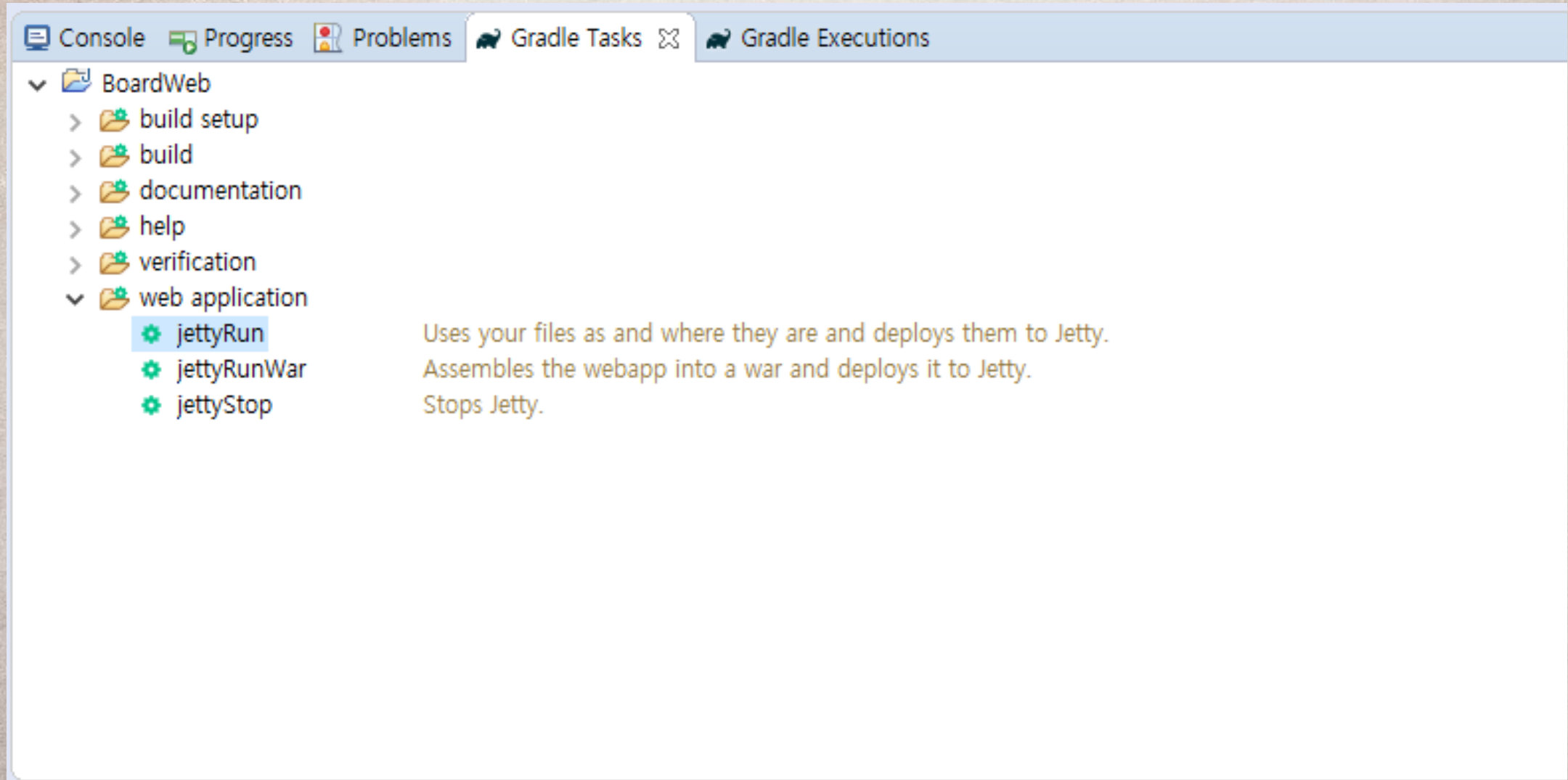
build.gradle - jetty server 추가

```
build.gradle applicationContext.xml JDBCUtil.java index.html web.xml DispatcherServlet.java
1 apply plugin: 'java'
2 apply plugin: 'jetty'
3
4 compileJava.options.encoding = 'UTF-8'
5
6 // In this section you declare where to find the dependencies of your project
7 repositories {
8     mavenCentral()
9 }
10
11 jettyRun{
12     httpPort = 8100
13 }
14
15 dependencies {
16     compile 'org.springframework:spring-context:4.3.8.RELEASE'
17     compile 'com.h2database:h2:1.4.195'
18     compile 'org.aspectj:aspectjweaver:1.8.8'
19     compile 'org.apache.commons:commons-dbcp2:2.1'
20     compile 'org.springframework:spring-jdbc:4.3.8.RELEASE'
21     compile 'org.springframework:spring-webmvc:4.3.8.RELEASE'
22     compile 'javax.servlet:javax.servlet-api:3.1.0'
23 }
24
25
```

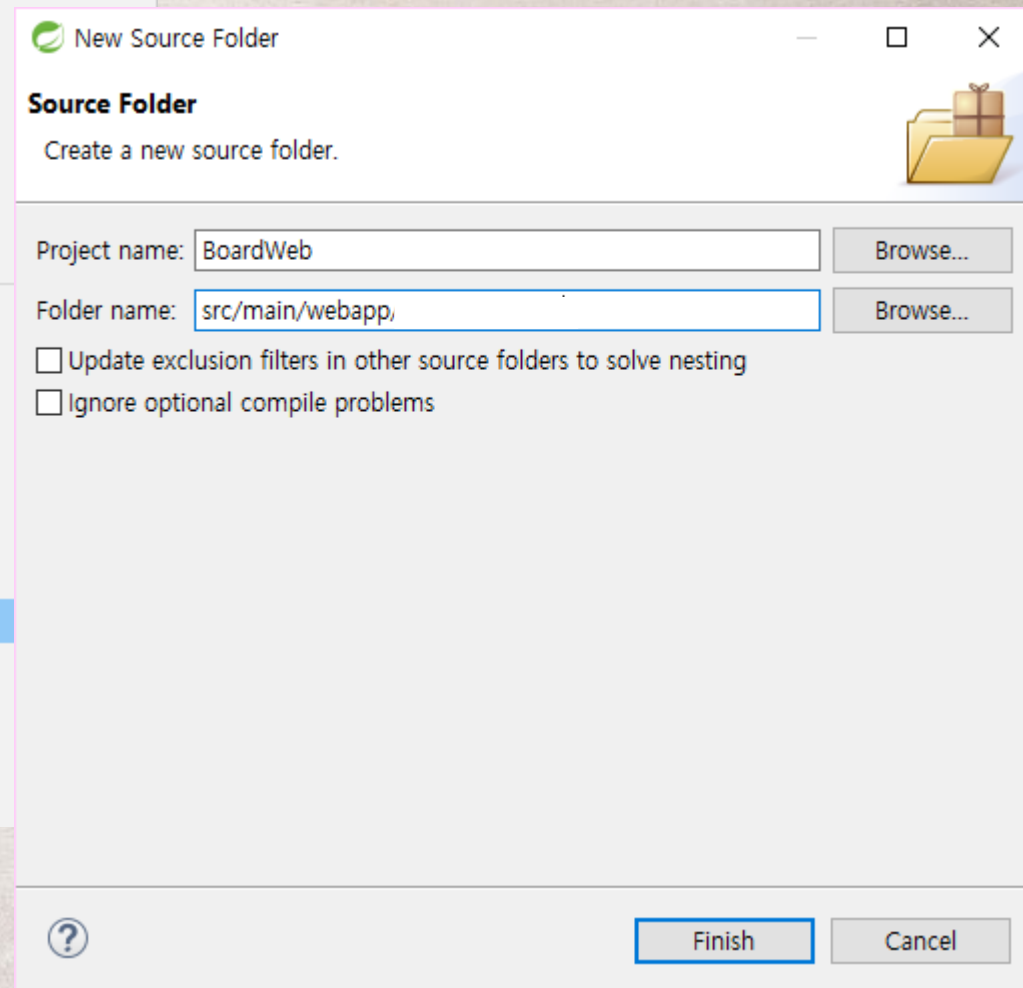
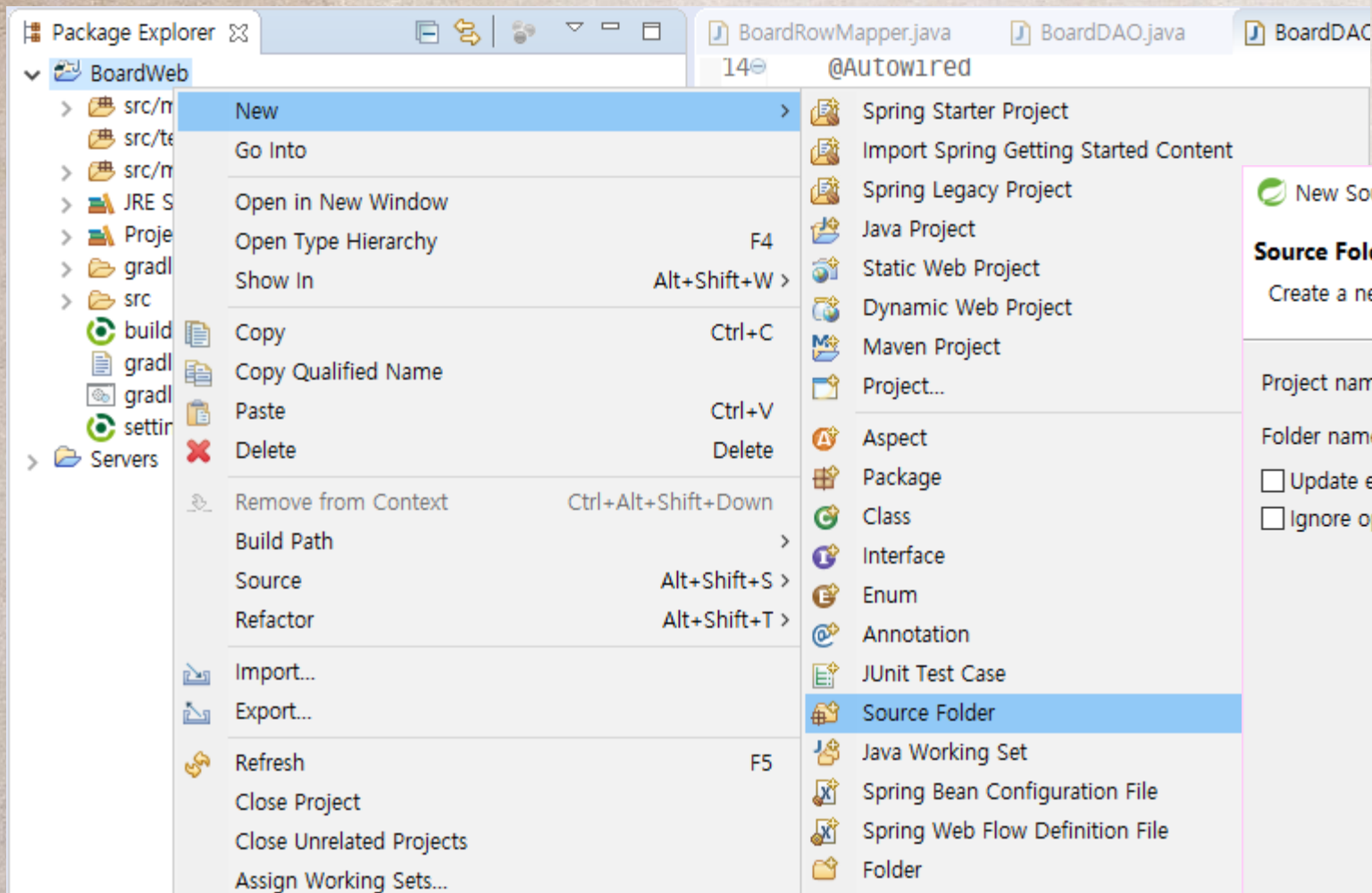
스프링 mvc 및 servlet 관련

Gradle tasks에 추가된 jetty web server

- Gradle Tasks 창을 F5 버튼을 눌러 리플래쉬 하면 나타남

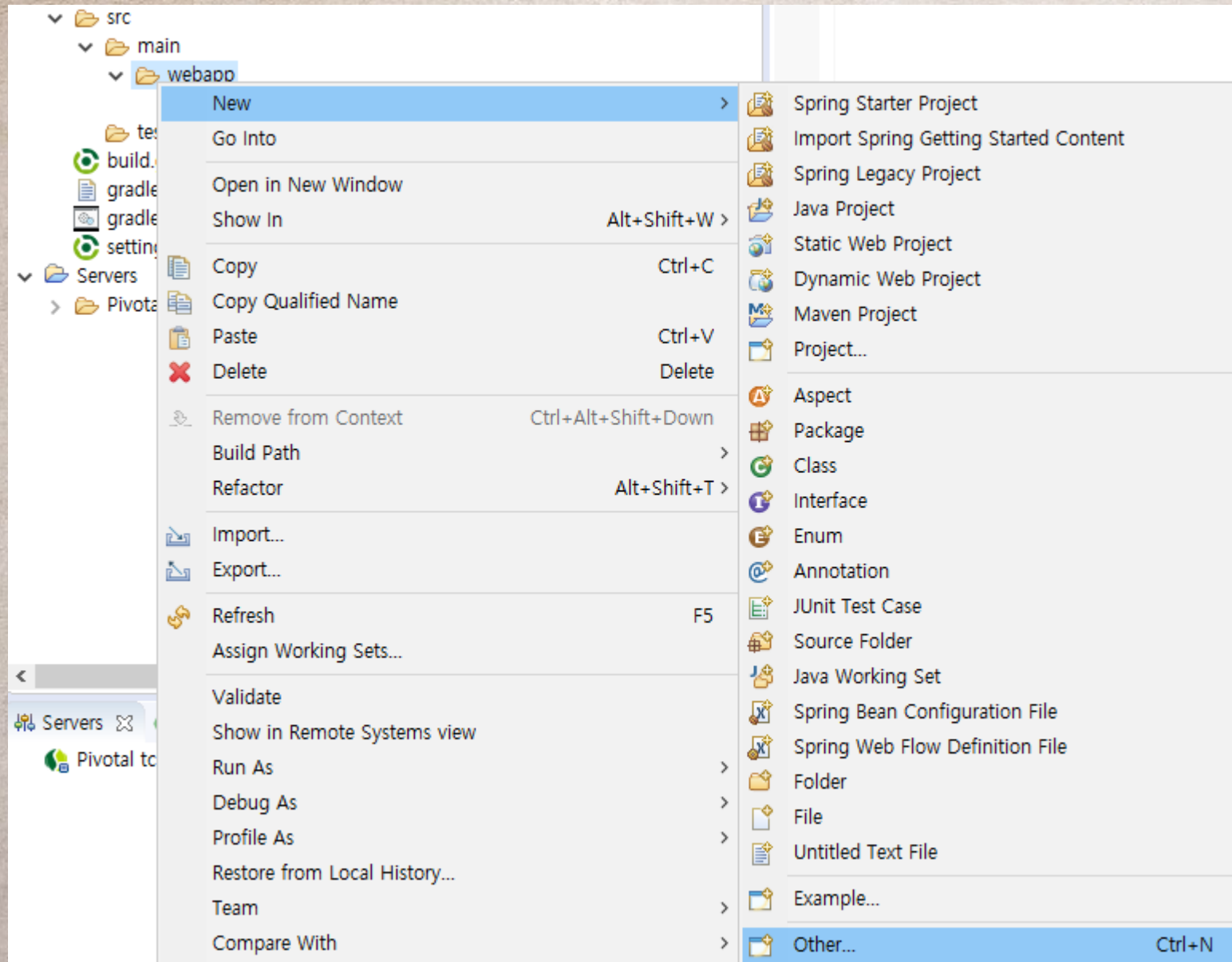


webapp 및 jsp 파일 소스 폴더 생성

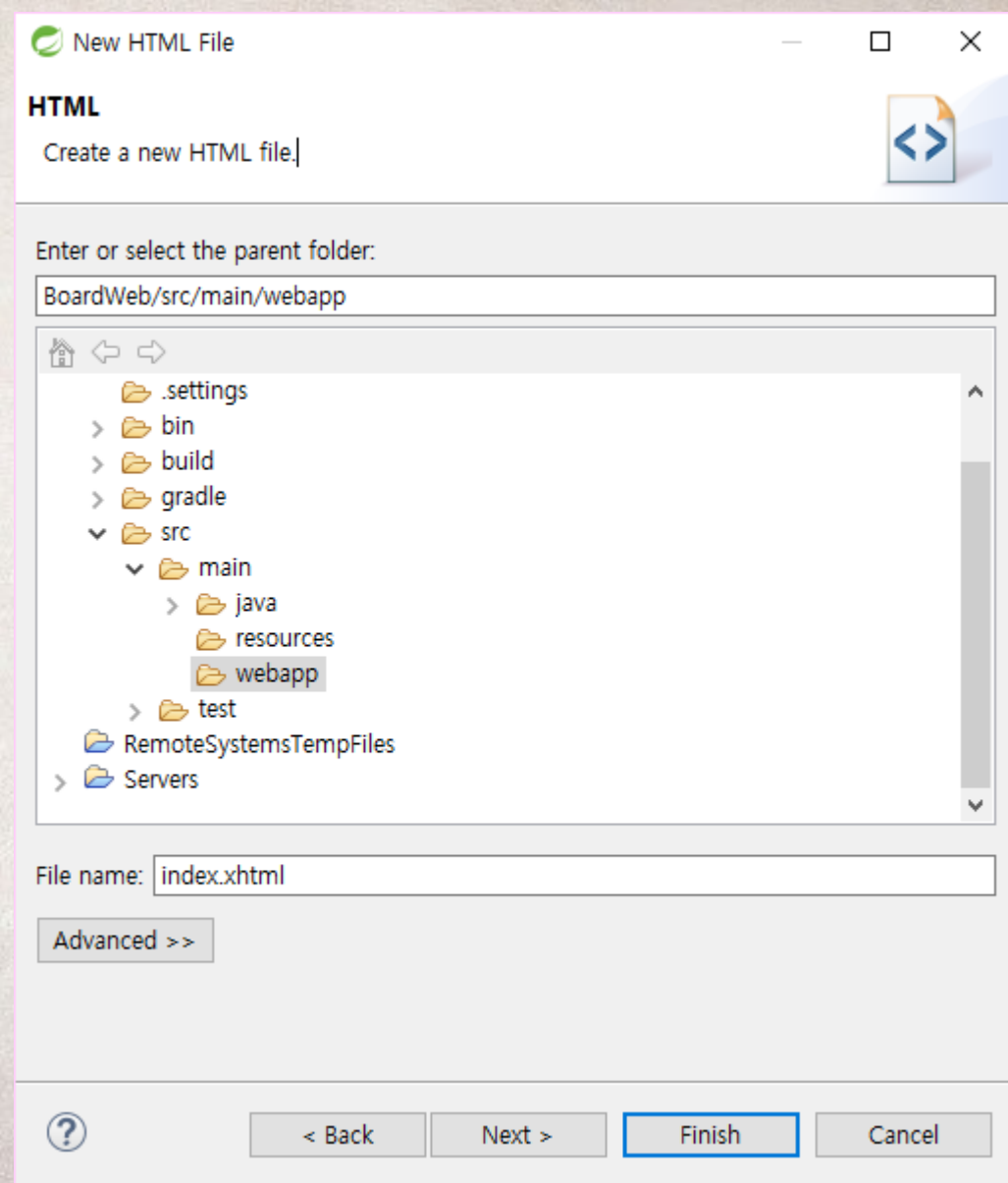
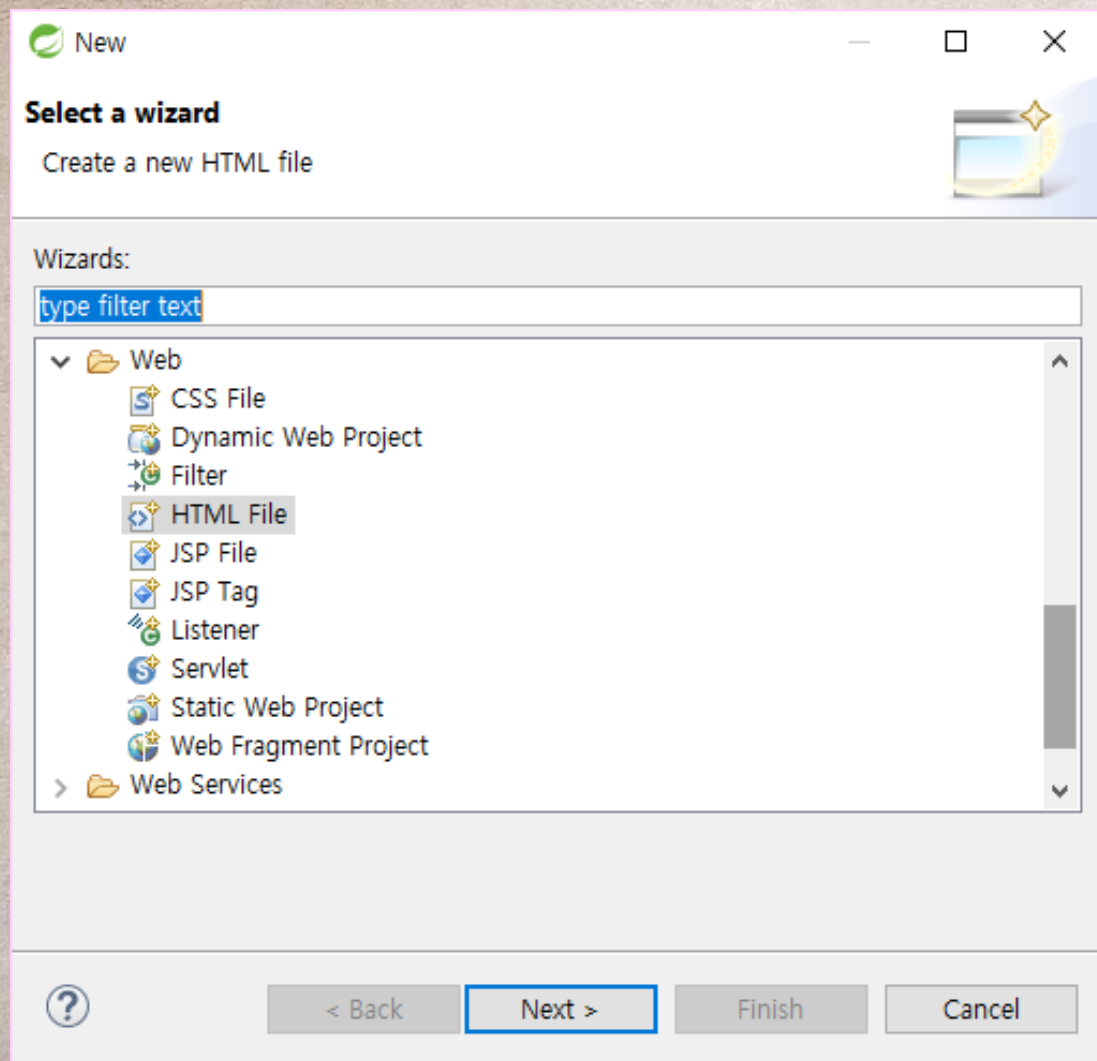


Folder name: src/main/webapp

src/main/webapp/index.html 생성



WEB-INF/index.html 생성



build.gradle applicationContext.xml JDBCUtil.java *index.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 게시판 시작 페이지
9 </body>
10 </html>
```

11

12

ConsoleProgressProblemsGradle TasksGradle Executions

BoardWeb

- build setup
- build
- documentation
- help
- verification
- web application
 - jettyRun
 - jettyRunWar
 - jettyStop

Uses your files as and where they are and deploys them to Jetty.

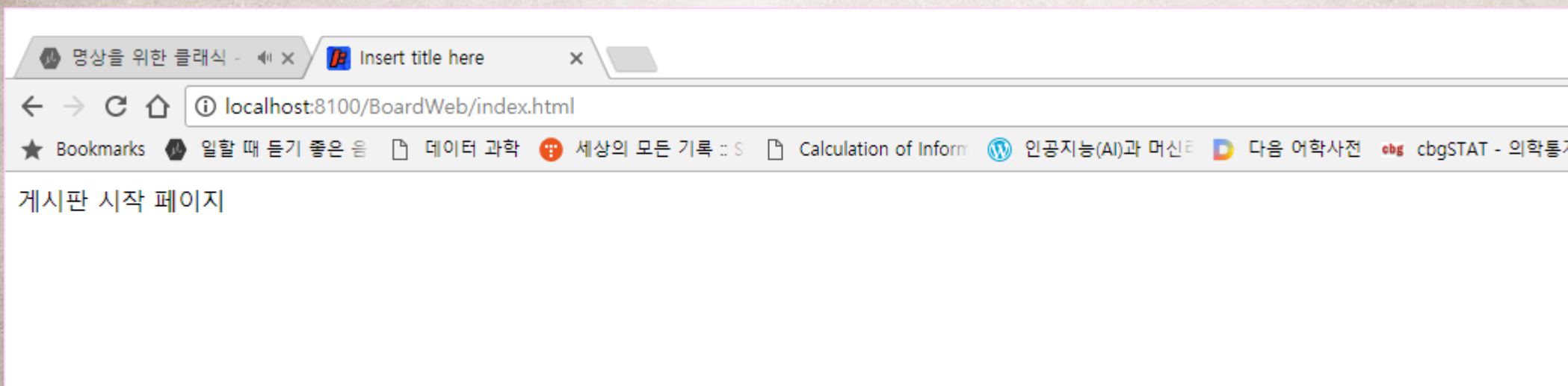
Assembles the webapp into a war and deploys it to Jetty.

Stops Jetty.

더블클릭


```
BoardWeb - jettyRun [Gradle Project] jettyRun in D:\workspace\java\2017_1\BoardWeb (2017. 9. 27 오후 2:04:57)
Working Directory: D:\workspace\java\2017_1\BoardWeb
Gradle User Home: C:\Users\Kyungtae\.gradle
Gradle Distribution: Local installation at C:\gradle-3.3
Gradle Version: 3.3
Java Home: C:\Program Files\Java\jdk1.8.0_144
JVM Arguments: None
Program Arguments: None
Gradle Tasks: jettyRun

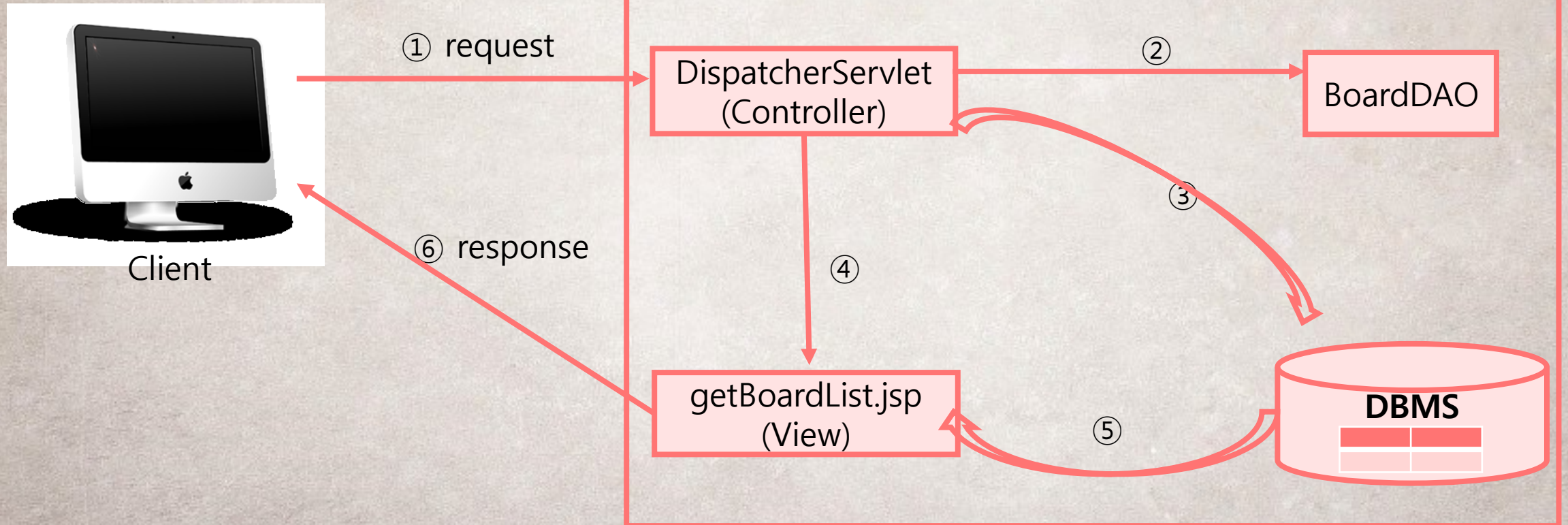
The Jetty plugin has been deprecated and is scheduled to be removed in Gradle 4.0. Consider using the Gretty (https://github.com/akhil
at build_asxoyenton2m5od0zoiqymdw1.run(D:\workspace\java\2017_1\BoardWeb\build.gradle:2)
:compileJava UP-TO-DATE
:processResources UP-TO-DATE
:classes UP-TO-DATE
:jettyRun
```



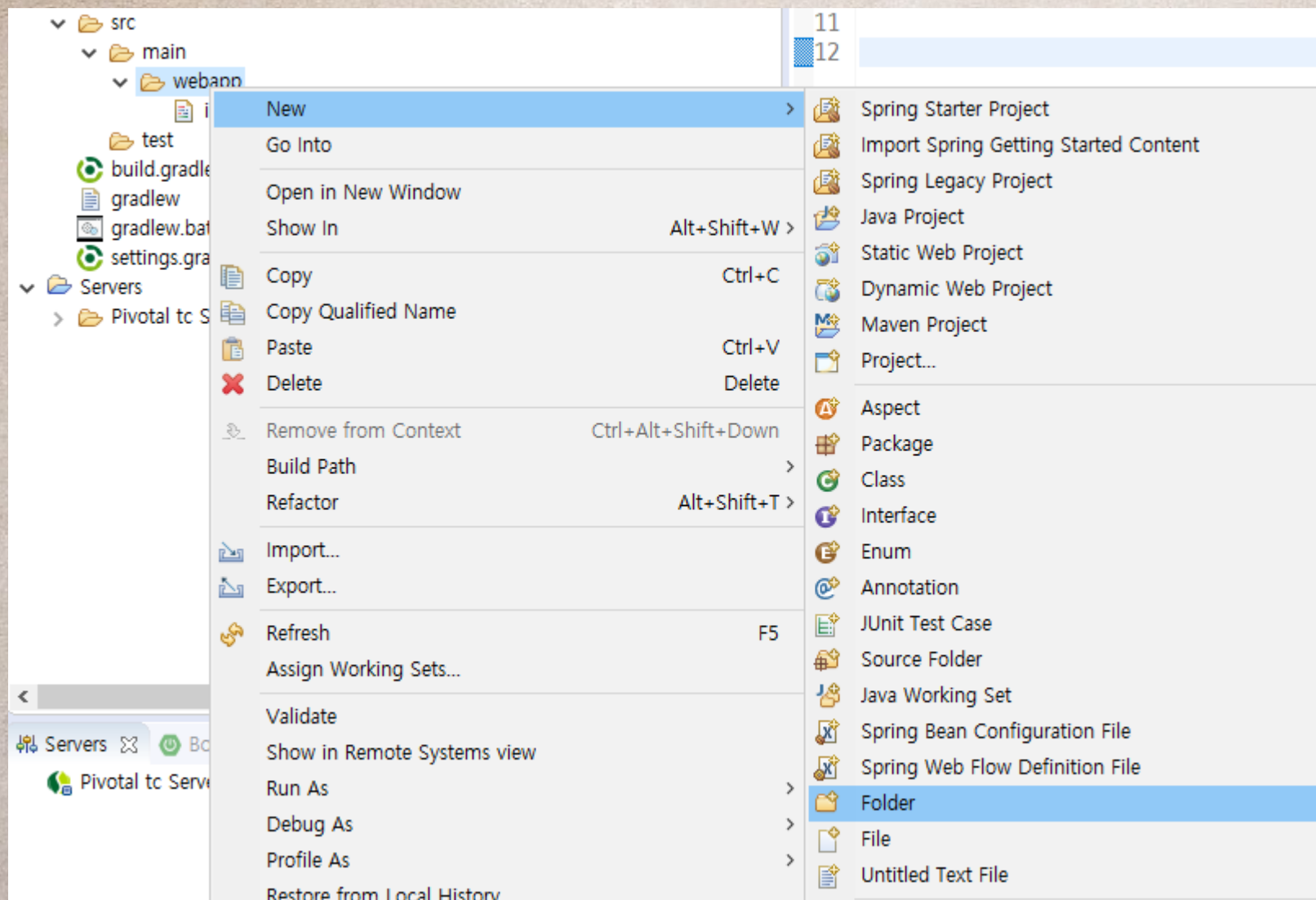
게시판 모델2 작업

게시판 목록 검색 실행 순서

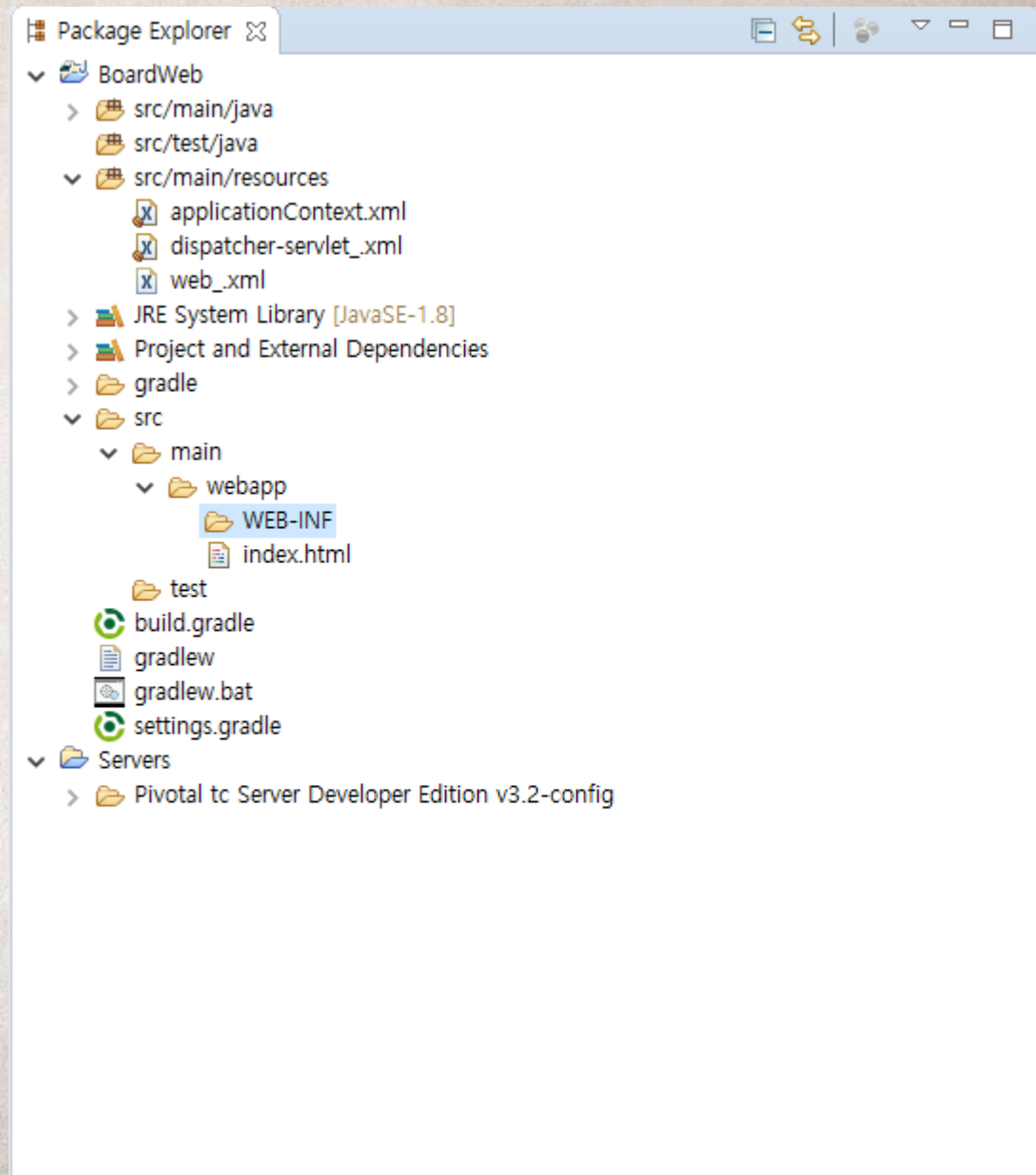
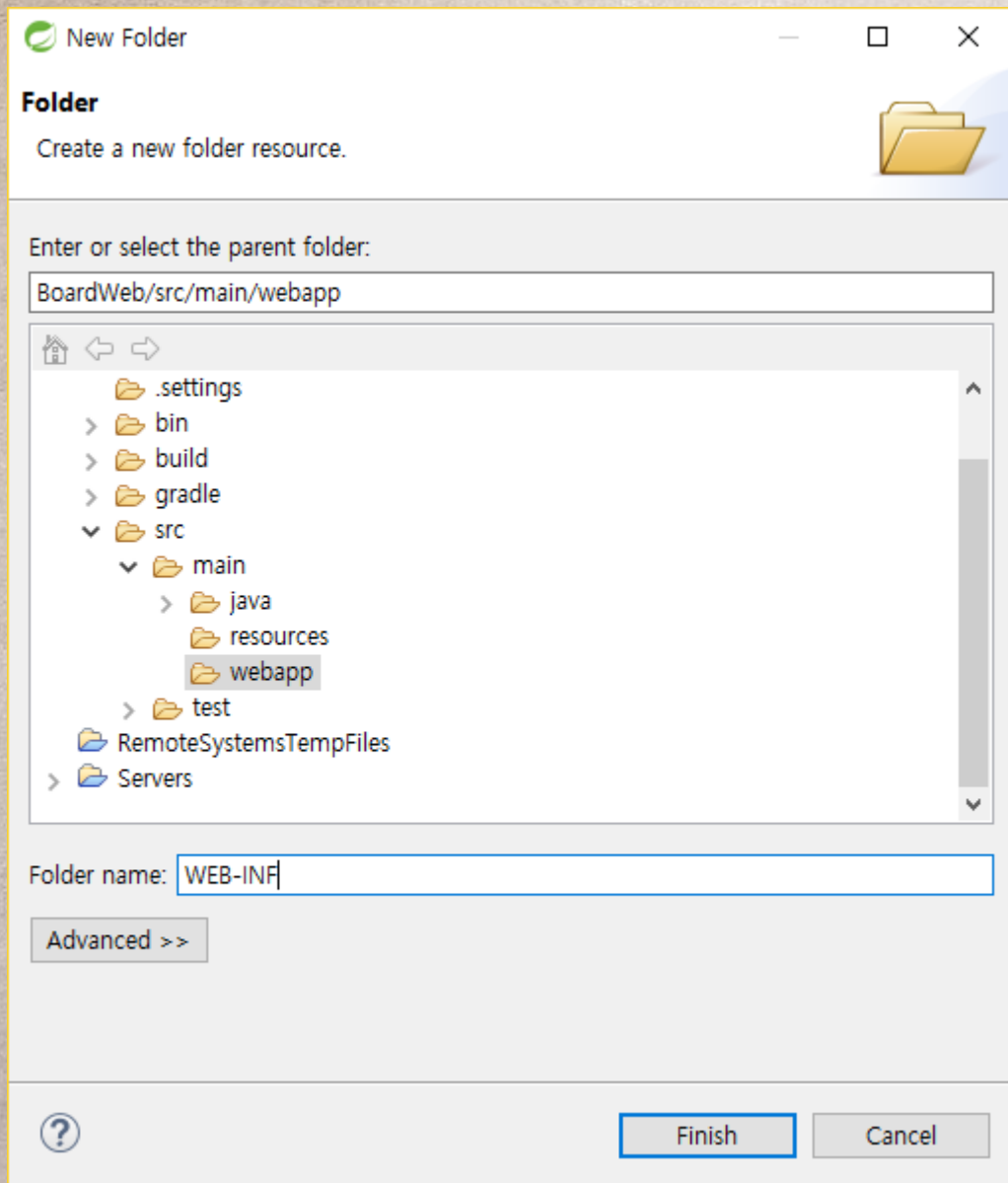
Model 2 아키텍처 구조



WEB-INF 폴더와 web.xml 파일 생성



WEB-INF 폴더 생성



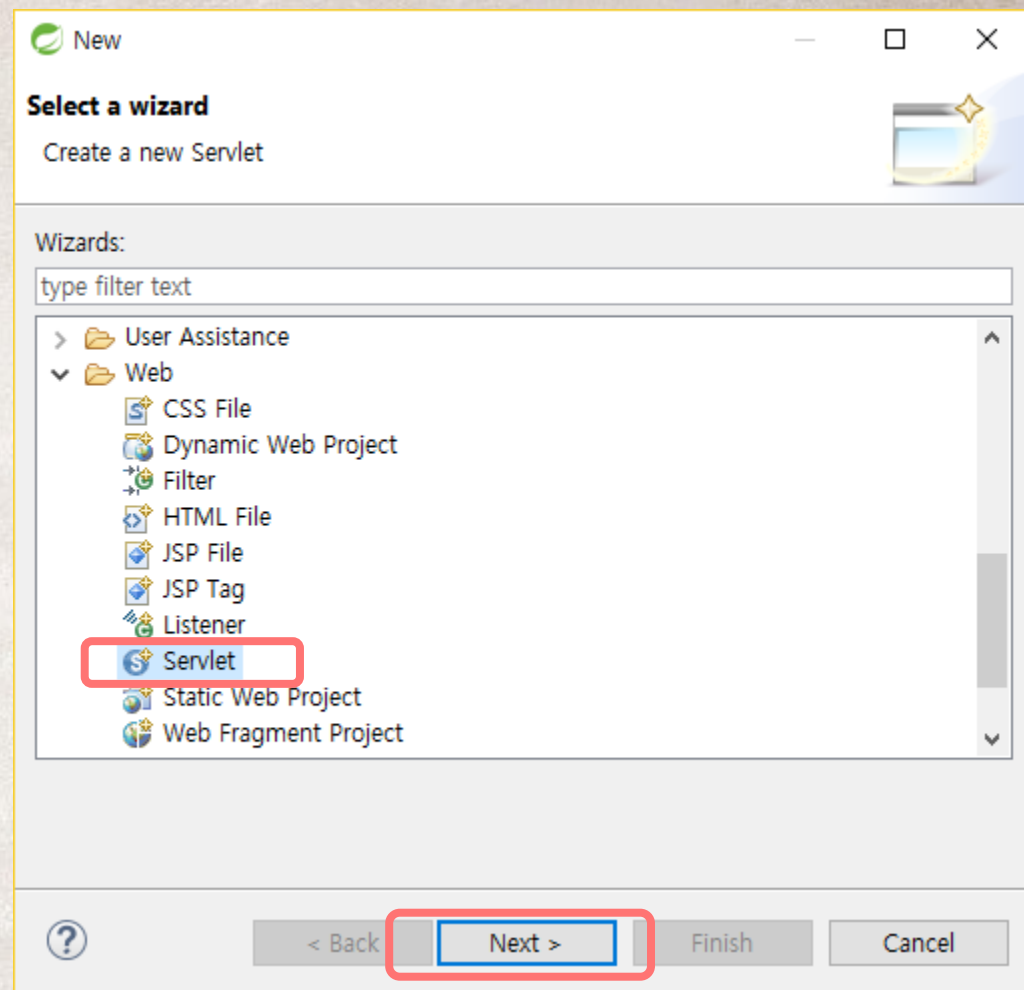
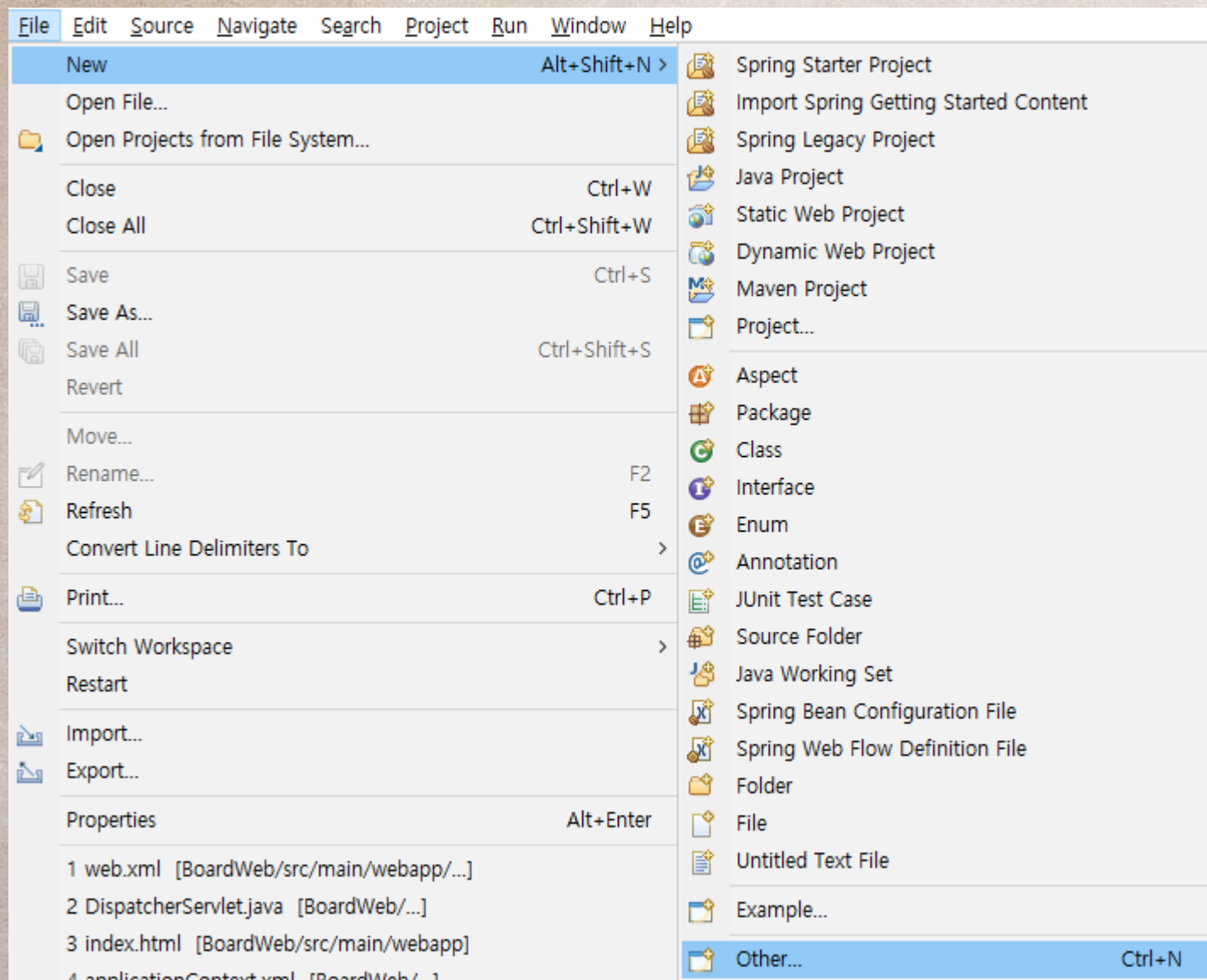
WEB-INF/web.xml 파일 생성

The screenshot shows an IDE interface with the Package Explorer on the left and the Source editor on the right. The Package Explorer shows a project named 'BoardWeb' with a directory structure including 'src/main/resources' and 'src/main/webapp'. The 'src/main/webapp' directory contains a 'WEB-INF' subdirectory, which is currently selected. The 'web.xml' file is highlighted within the 'WEB-INF' directory. The Source editor displays the content of the 'web.xml' file, which is an XML document defining a web application. The XML content is as follows:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5         version="3.1">
6
7     <welcome-file-list>
8         <welcome-file>index.html</welcome-file>
9         <welcome-file>index.htm</welcome-file>
10        <welcome-file>index.jsp</welcome-file>
11    </welcome-file-list>
12 </web-app>
13
14
```

The Source editor also shows tabs for 'build.gradle', 'applicationContext.xml', 'JDBCUtil.java', 'index.html', and 'web.xml'. The 'Design' and 'Source' tabs are visible at the bottom of the editor.

Controller 구현 - DispatcherServlet 구현



Create Servlet

Specify class file destination.

Project: BoardWeb

Source folder: /BoardWeb/src/main/java

Java package: kr.ac.inje.comsi.view.controller

Class name: DispatcherServlet

Superclass: javax.servlet.http.HttpServlet

☐ Use an existing Servlet class or JSP

Class name: DispatcherServlet

< Back Next > Finish Cancel

Create Servlet

Enter servlet deployment descriptor specific information.

Name: action

Description:

Initialization parameters:

Name	Value	Description
------	-------	-------------

URL mappings:

/action

Edit...

< Back Next > Finish Cancel

Create Servlet


Enter servlet deployment descriptor specific information.



Name:

Description:

Initialization parameters:

Name
<div><div> URL Mappings</div><div><div>Pattern: <input type="text" value="*.do"/></div><div><input type="button" value="OK"/> <input type="button" value="Cancel"/></div></div></div>

URL mappings:

/action



< Back

Next >

변경된 web.xml

```
build.gradle  applicationContext.xml  JDBCUtil.java  index.html  web.xml  DispatcherServlet.java

1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4      xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5      version="3.1">
6      <servlet>
7          <description>
8          </description>
9          <display-name>action</display-name>
10         <servlet-name>action</servlet-name>
11         <servlet-class>kr.ac.inje.comsi.view.controller.DispatcherServlet</servlet-class>
12     </servlet>
13     <servlet-mapping>
14         <servlet-name>action</servlet-name>
15         <url-pattern>*.do</url-pattern>
16     </servlet-mapping>
17
18     <welcome-file-list>
19         <welcome-file>index.html</welcome-file>
20         <welcome-file>index.htm</welcome-file>
21         <welcome-file>index.jsp</welcome-file>
22     </welcome-file-list>
23 </web-app>
24
25
```


DispatcherServlet.java 클래스

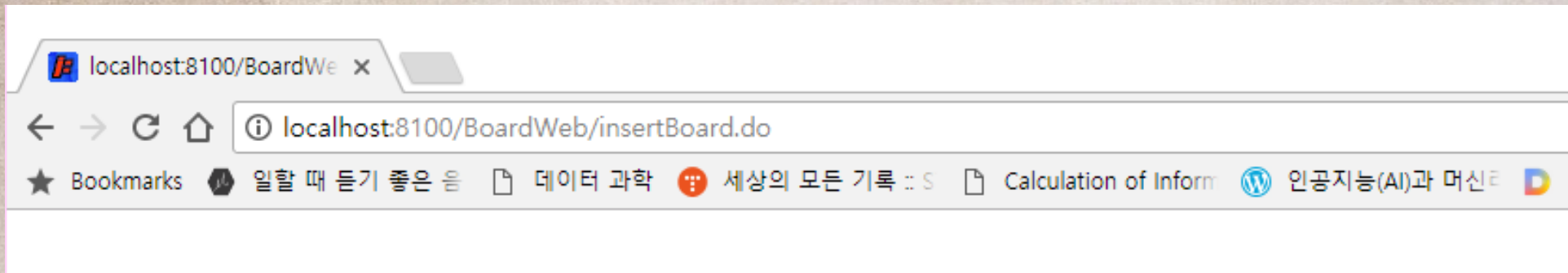
```
web.xml DispatcherServlet.java
1 package kr.ac.inje.comsi.view.controller;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.http.HttpServlet;
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9
10 /**
11  * Servlet implementation class DispatcherServlet
12  */
13 public class DispatcherServlet extends HttpServlet {
14     private static final long serialVersionUID = 1L;
15
16     /**
17      * Default constructor.
18      */
19     public DispatcherServlet() {
20         // TODO Auto-generated constructor stub
21     }
22
23     /**
24      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
25      */
26     protected void doGet(HttpServletRequest request, HttpServletResponse response)
27         throws ServletException, IOException {
28         // TODO Auto-generated method stub
29         process(request, response);
30     }
31 }
```

```

32- /**
33-  * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
34-  */
35- protected void doPost(HttpServletRequest request, HttpServletResponse response)
36-     throws ServletException, IOException {
37-     // TODO Auto-generated method stub
38-     request.setCharacterEncoding("UTF-8");
39-     process(request, response);
40- }
41-
42- private void process(HttpServletRequest request, HttpServletResponse response)
43-     throws IOException{
44-     // 1. 클라이언트의 요청 path 정보를 추출한다.
45-     String uri = request.getRequestURI();
46-     String path = uri.substring(uri.lastIndexOf("/"));
47-     System.out.println(path);
48-
49-     // 2. 클라이언트의 요청 path에 따라 적절히 분기처리 한다.
50-     if(path.equals("/login.do")){
51-         System.out.println("로그인 처리");
52-     }else if(path.equals("/logout.do")){
53-         System.out.println("로그아웃 처리");
54-     }else if(path.equals("/insertBoard.do")){
55-         System.out.println("글 등록 처리");
56-     }else if(path.equals("/updateBoard.do")){
57-         System.out.println("글 수정 처리");
58-     }else if(path.equals("/deleteBoard.do")){
59-         System.out.println("글 삭제 처리");
60-     }else if(path.equals("/getBoard.do")){
61-         System.out.println("글 상세 조회 처리");
62-     }else if(path.equals("/getBoardList.do")){
63-         System.out.println("글 목록 검색 처리");
64-     }
65- }
66-
67- }
68-

```


Controller 테스트



BoardWeb - jettyRun [Gradle Project] jettyRun in D:\workspace\java\2017_1\BoardWeb (2017. 9. 28 오후 7:40:13)

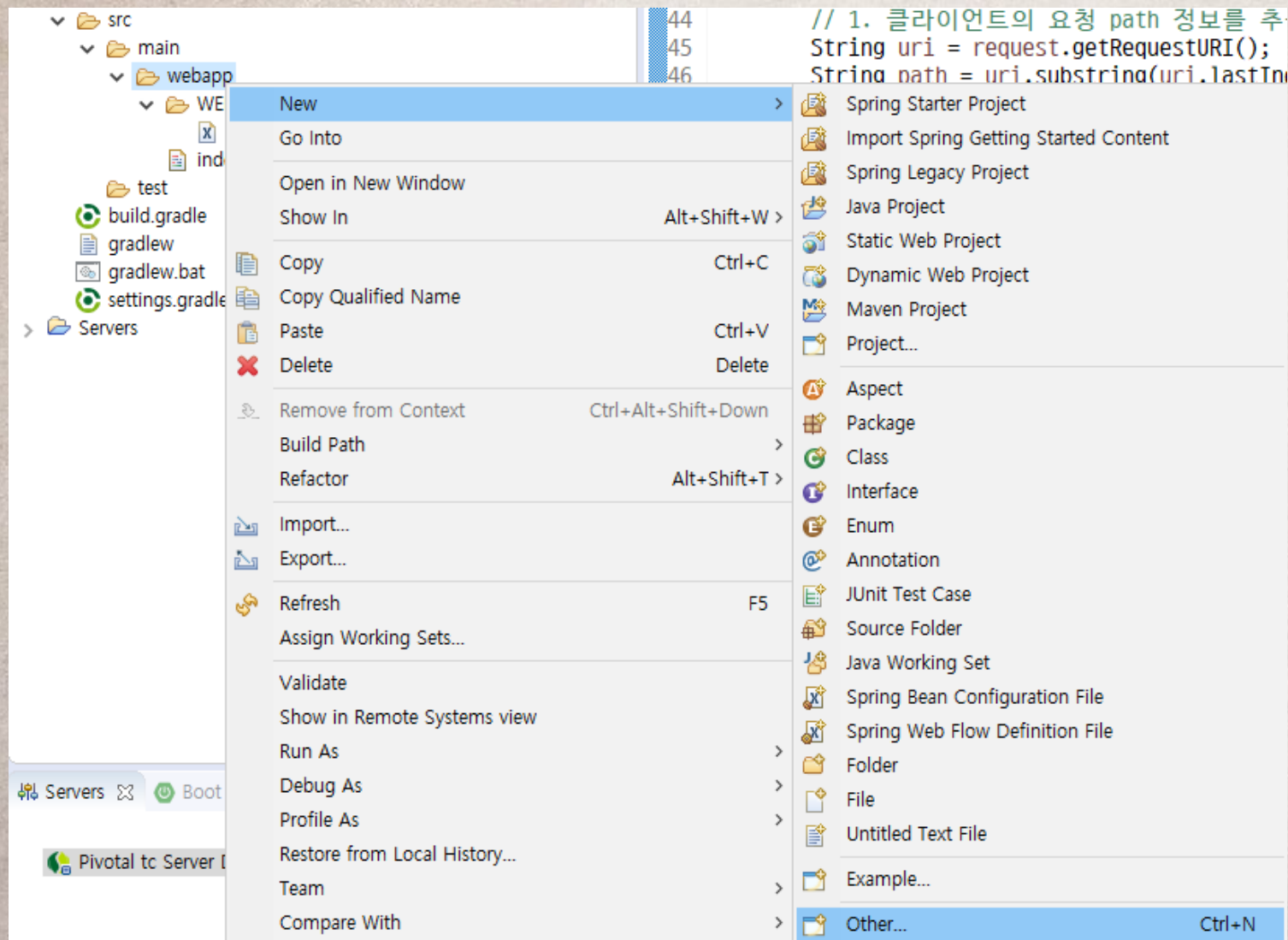
Gradle Tasks: jettyRun

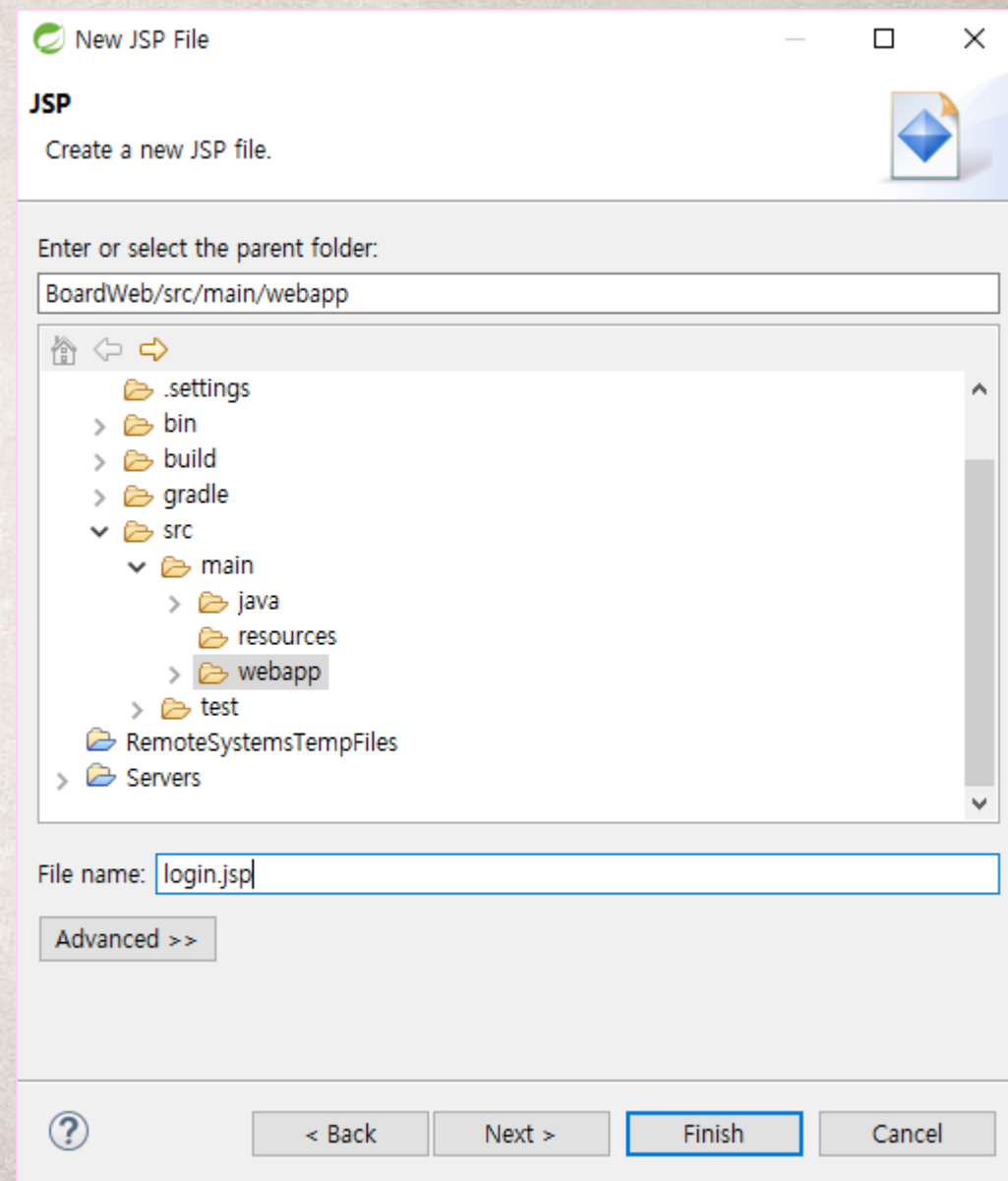
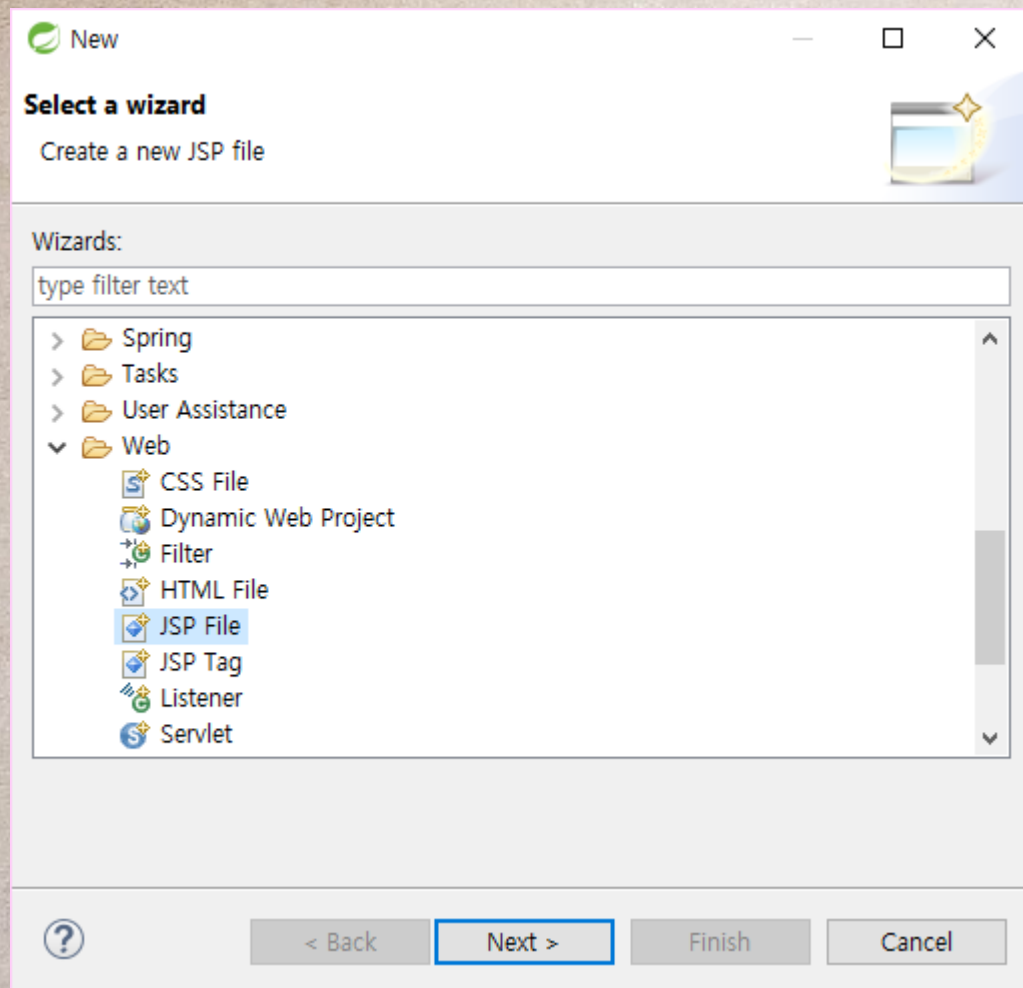
The Jetty plugin has been deprecated and is scheduled to be removed in Gradle 4.0. Consider using the Gretty (<https://github.com/gradle-plugins/gretty>) at build_asxoyenton2m5od0zoiqymdw1.run(D:\workspace\java\2017_1\BoardWeb\build.gradle:2)

:compileJava UP-TO-DATE
:processResources UP-TO-DATE
:classes UP-TO-DATE
:jettyRun
/login.do
로그인 처리
/insertBoard.do
글 등록 처리

JSP 파일(View 구현-login.jsp)

- src/main/webapp 폴더에 등록해야 함.





login.jsp

```
web.xml *DispatcherServlet.java login.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>로그인</title>
8 </head>
9 <body>
10 <center>
11 <h1>로그인</h1>
12 <hr>
13 <form action="login.do" method="post">
14 <table border="1" cellpadding="0" cellspacing="0">
15   <tr>
16     <td bgcolor="orange">아이디</td>
17     <td><input type="text" name="id"/></td>
18   </tr>
19   <tr>
20     <td bgcolor="orange">비밀번호</td>
21     <td><input type="password" name="password" /></td>
22   </tr>
23   <tr>
24     <td colspan="2" align="center"><input type="submit" value="로그인"/></td>
25   </tr>
26 </table>
27 </form>
28 <hr>
29 </center>
30 </body>
31 </html>
```


http://localhost:8100/BoardWeb/login.jsp

로그인

← → ↻ 🏠 ⓘ localhost:8100/BoardWeb/login.jsp ☆ ⋮

★ Bookmarks 📁 일할 때 듣기 좋은 음 📄 데이터 과학 🗄️ 세상의 모든 기록 :: S 📄 Calculation of Inform 🌐 인공지능(AI)과 머신러 📖 다음 어학사전 »

로그인

아이디	<input type="text"/>
비밀번호	<input type="password"/>
<input type="button" value="로그인"/>	<input type="button" value=""/>

➔ 아이디와 비밀번호 입력 후 로그인

DispatcherServlet 클래스의 “/login.do” 호출

```
52 // 2. 클라이언트의 요청 path에 따라 적절히 분기처리 한다.
53 if(path.equals("/login.do")){
54     System.out.println("로그인 처리");
55     // 1. 사용자 입력 정보 추출
56     String id = request.getParameter("id");
57     String password = request.getParameter("password");
58
59     // 2. DB 연동 처리
60     UserVO vo = new UserVO();
61     vo.setId(id);
62     vo.setPassword(password);
63
64     UserDAO userDAO = new UserDAO();
65     UserVO user = userDAO.getUser(vo);
66
67     // 3. 화면 네비게이션
68     if(user != null){
69         response.sendRedirect("getBoardList.do");
70     }else{
71         response.sendRedirect("login.jsp");
72     }
73 }else if(path.equals("/logout.do")){
74     System.out.println("로그아웃 처리");
75 }else if(path.equals("/insertBoard.do")){
76     System.out.println("글 등록 처리");
```

추가

DispatcherServlet 클래스의 “/getBoardList.do” 호출

```
88     }else if(path.equals("/getBoardList.do")){
89         System.out.println("글 목록 검색 처리");
90         // 1. 사용자 입력 정보 추출(검색 기능은 나중에 구현)
91         // 2. DB 연동 처리
92         BoardVO vo = new BoardVO();
93         BoardDAO boardDAO = new BoardDAO();
94         List<BoardVO> boardList = boardDAO.getBoardList(vo);
95
96         // 3. 검색 결과를 세션에 저장하고 목록 화면으로 이동한다.
97         HttpSession session = request.getSession();
98         session.setAttribute("boardList", boardList);
99         response.sendRedirect("getBoardList.jsp");
100     }
101 }
102
```

추가

getBoardList.jsp

```
web.xml DispatcherServlet.java login.jsp applicationContext.xml getBoardList.jsp
1 <%@ page import="java.util.List" %>
2 <%@ page import="kr.ac.inje.comsi.board.BoardVO" %>
3 <%@ page language="java" contentType="text/html; charset=UTF-8"
4   pageEncoding="UTF-8"%>
5 <%
6   // 세션에 저장된 글 목록을 꺼낸다.
7   List<BoardVO> boardList = (List) session.getAttribute("boardList");
8 %>
9 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <title>글 목록</title>
14 </head>
15 <body>
16 <center>
17   <h1>글 목록</h1>
18   <h3>환영합니다. <a href="logout.do">Logout</a></h3>
19
20   <!-- 검색시작 -->
21   <form action="getBoardList.jsp" method="post">
22   <table border="1" cellpadding="0" cellspacing="0" width="700">
23   <tr>
24     <td align="right">
25       <select name="searchCondition">
26         <option value="TITLE">제목</option>
27         <option value="CONTENT">내용</option>
28       </select>
29       <input name="searchKeyword" type="text"/>
30       <input type="submit" value="검색"/>
31     </td>
32   </tr>
33   </table>
34   </form>
35   <!-- 검색종료 -->
```



```

36
37<Ⓜ <table border="1" cellpadding="0" cellspacing="0" width="700">
38<Ⓜ <tr>
39    <th bgcolor="orange" width="100">번호</th>
40    <th bgcolor="orange" width="200">제목</th>
41    <th bgcolor="orange" width="150">작성자</th>
42    <th bgcolor="orange" width="150">등록일</th>
43    <th bgcolor="orange" width="100">조회수</th>
44</tr>
45<Ⓜ <% for(BoardVO board : boardList) { %>
46<Ⓜ <tr>
47    <td align="center"><%= board.getSeq() %></td>
48    <td align="left"><a href="getBoard.do?seq=<%= board.getSeq() %>"><%= board.getTitle() %></a></td>
49    <td align="center"><%= board.getWriter() %></td>
50    <td align="center"><%= board.getRegDate() %></td>
51    <td align="center"><%= board.getCnt() %></td>
52</tr>
53<Ⓜ <% } %>
54
55</table>
56<br>
57<a href="insertBoard.do">새글 등록</a>
58</center>
59</body>
60</html>
61
62

```

로그인 후 글 목록 검색하기

로그인

H2 콘솔

← → ↻ 🏠 ⓘ localhost:8100/BoardWeb/login.jsp ☆ ⋮

★ Bookmarks 🗄️ 일할 때 듣기 좋은 음 📄 데이터 과학 🗣️ 세상의 모든 기록 :: S 📄 Calculation of Inform 🌐 인공지능(AI)과 머신러닝 📖 다음 어학사전 📄 cbg cbgSTAT - 의학통계 »

로그인

아이디

test

비밀번호

.....

로그인

글 목록

환영합니다. [Logout](#)

<div> <div>제목 ▼</div> <div></div> <div>검색</div> </div>				
번호	제목	작성자	등록일	조회수
16	JDBC테스트	홍길동	2017-09-27	0
15	JDBC테스트	홍길동	2017-09-26	0
14	JDBC테스트	홍길동	2017-09-25	0
13	JDBC테스트	홍길동	2017-09-25	0
12	AOP실행	홍길동	2017-09-13	0
11	AOP실행	홍길동	2017-09-13	0
10	AOP실행	홍길동	2017-09-13	0
9	AOP실행	홍길동	2017-09-13	0
8	AOP실행	홍길동	2017-09-13	0
7	AOP실행	홍길동	2017-09-09	0
6	AOP실행	김길동2	2017-09-09	0
5	AOP실행	김길동2	2017-09-09	0
4	AOP실행	김길동	2017-09-09	0
3	AOP실행	김길동	2017-09-09	0
2	AOP실행	김길동	2017-09-09	0
1	가입인사	관리자	2017-09-09	0

[새글 등록](#)

게시판 목록 검색 실행 순서

- ① DispatcherServlet이 클라이언트의 “/getBoardList.do” 요청을 받으면
- ② DispatcherServlet은 BoardDAO 객체를 이용하여 글 목록을 검색한다.
- ③ 검색된 글 목록을 세션에 등록하고
- ④ getBoardList.jsp 화면을 요청하면,
- ⑤ getBoardList.jsp는 세션에 저장된 글 목록을 꺼내어 목록 화면을 구성한다.
- ⑥ 마지막으로 이 응답 화면이 브라우저에 전송된다.