



박 경 태

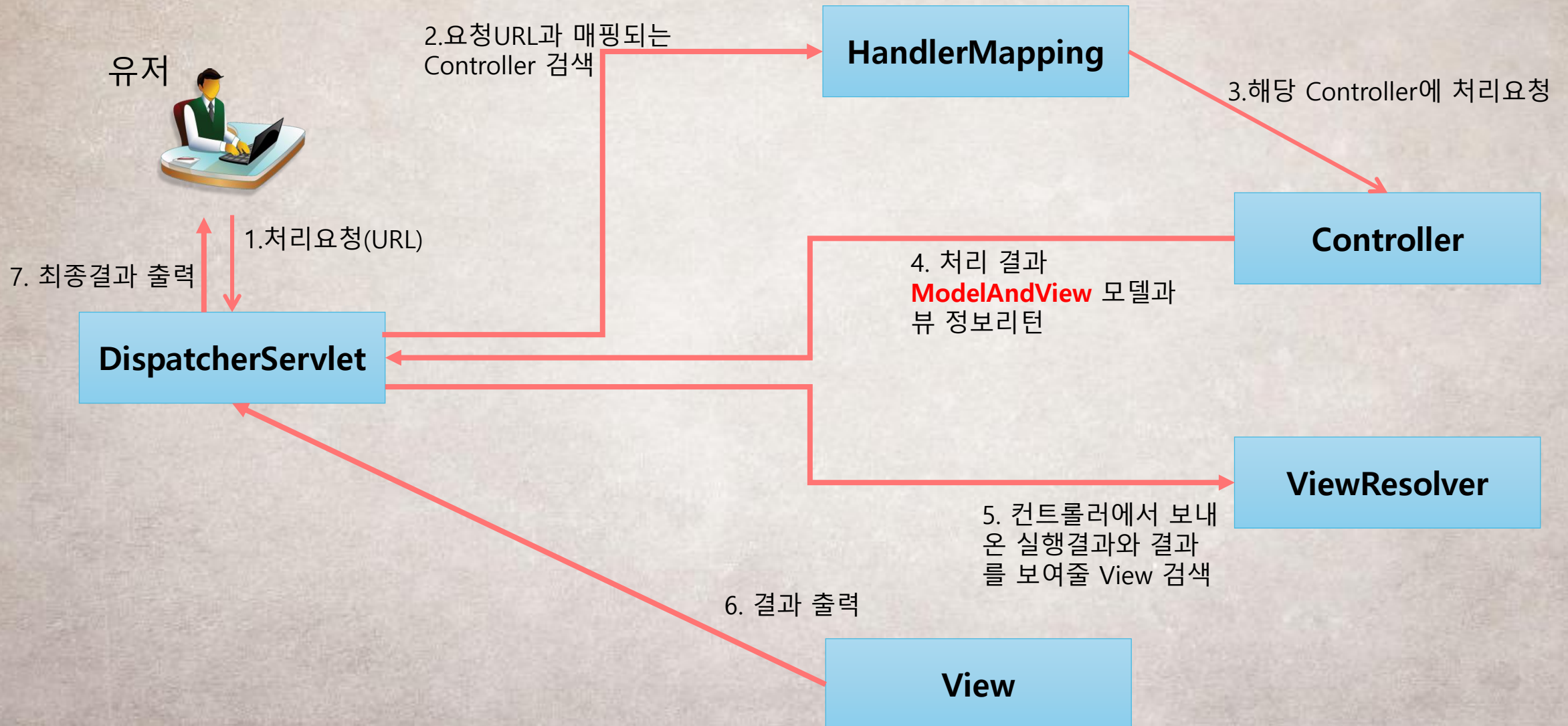
comsi.java@gmail.com

고급 자바 프로그래밍
: STS를 이용한 Spring 프로그래밍

5. SpringMVC 구조

5.1 스프링 MVC의 수행 흐름

스프링 MVC의 클라이언트 요청 처리 과정



스프링 MVC를 이용한 웹어플리케이션

- 스프링 MVC의 주요 구성 요소

구성요소	설명
DispatcherServlet	클라이언트의 요청을 전달 받는다. Controller에게 클라이언트의 요청을 전달하고, Controller가 리턴한 결과값을 View에 전달하여 알맞은 응답을 생성하도록 한다.
HandlerMapping	클라이언트의 요청 URL을 어떤 Controller가 처리할지를 결정한다.
Controller	클라이언트의 요청을 처리한 뒤, 그 결과를 DispatcherServlet에 알려 준다.
ModelAndView	Controller가 처리한 결과 정보 및 뷰 선택에 필요한 정보를 담는다.
ViewResolver	Controller의 처리 결과를 생성할 View를 결정한다.
View	Controller의 처리 결과 화면을 생성한다.

5.2 DispatcherServlet 등록 및 스프링컨테이너 구동

5.2.1 DispatcherServlet 등록

- 우선 web.xml 파일을 수정하자.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://java.sun.com/xml/ns/javaee"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5                             http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
6
7     <servlet>
8         <display-name>DispatcherServlet</display-name>
9         <servlet-name>DispatcherServlet</servlet-name>
10        <!-- <servlet-class>kr.ac.inje.comsi.view.controller.DispatcherServlet</servlet-class> -->
11        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
12
13    </servlet>
14    <servlet-mapping>
15        <servlet-name>DispatcherServlet</servlet-name>
16        <url-pattern>*.do</url-pattern>
17    </servlet-mapping>
18 </web-app>
19
20
```

5.2.2 스프링 컨테이너 구동 → login.do 실행 후 오류

```
Console Progress Problems Markers
Pivot_tcServer - Pivotal tc Server v4.0 at localhost [Pivotal tc Server] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 9. 22. 오후 5:35:05)
9월 22, 2018 5:35:06 오후 org.apache.catalina.startup.Catalina load
정보: Initialization processed in 898 ms
9월 22, 2018 5:35:07 오후 org.apache.catalina.startup.Catalina start
정보: Server startup in 1077 ms
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'DispatcherServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'DispatcherServlet-serv
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/Dispat
ERROR: org.springframework.web.servlet.DispatcherServlet - Context initialization failed
org.springframework.beans.factory.BeanDefinitionStoreException: IOException parsing XML document from ServletContext resource [/WEB-INF/Dispatche
    at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions(XmlBeanDefinitionReader.java:344)
    at org.springframework.beans.factory.xml.XmlBeanDefinitionReader.loadBeanDefinitions(XmlBeanDefinitionReader.java:304)
    at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:181)
    at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:217)
    at org.springframework.beans.factory.support.AbstractBeanDefinitionReader.loadBeanDefinitions(AbstractBeanDefinitionReader.java:188)
    at org.springframework.web.context.support.XmlWebApplicationContext.loadBeanDefinitions(XmlWebApplicationContext.java:125)
    at org.springframework.web.context.support.XmlWebApplicationContext.loadBeanDefinitions(XmlWebApplicationContext.java:94)
    at org.springframework.context.support.AbstractRefreshableApplicationContext.refreshBeanFactory(AbstractRefreshableApplicationContext.jav
    at org.springframework.context.support.AbstractApplicationContext.obtainFreshBeanFactory(AbstractApplicationContext.java:614)
```

- 서버가 구동되면 DispatcherServlet이 구동되어 서버가 XmlWebApplicationContext라는 스프링 컨테이너가 구동된다.
- **Bean 정의 xml 파일 없어서...(박스 마지막 줄)**
 - DispatcherServlet-servlet.xml (기본 bean 설정 파일이름)

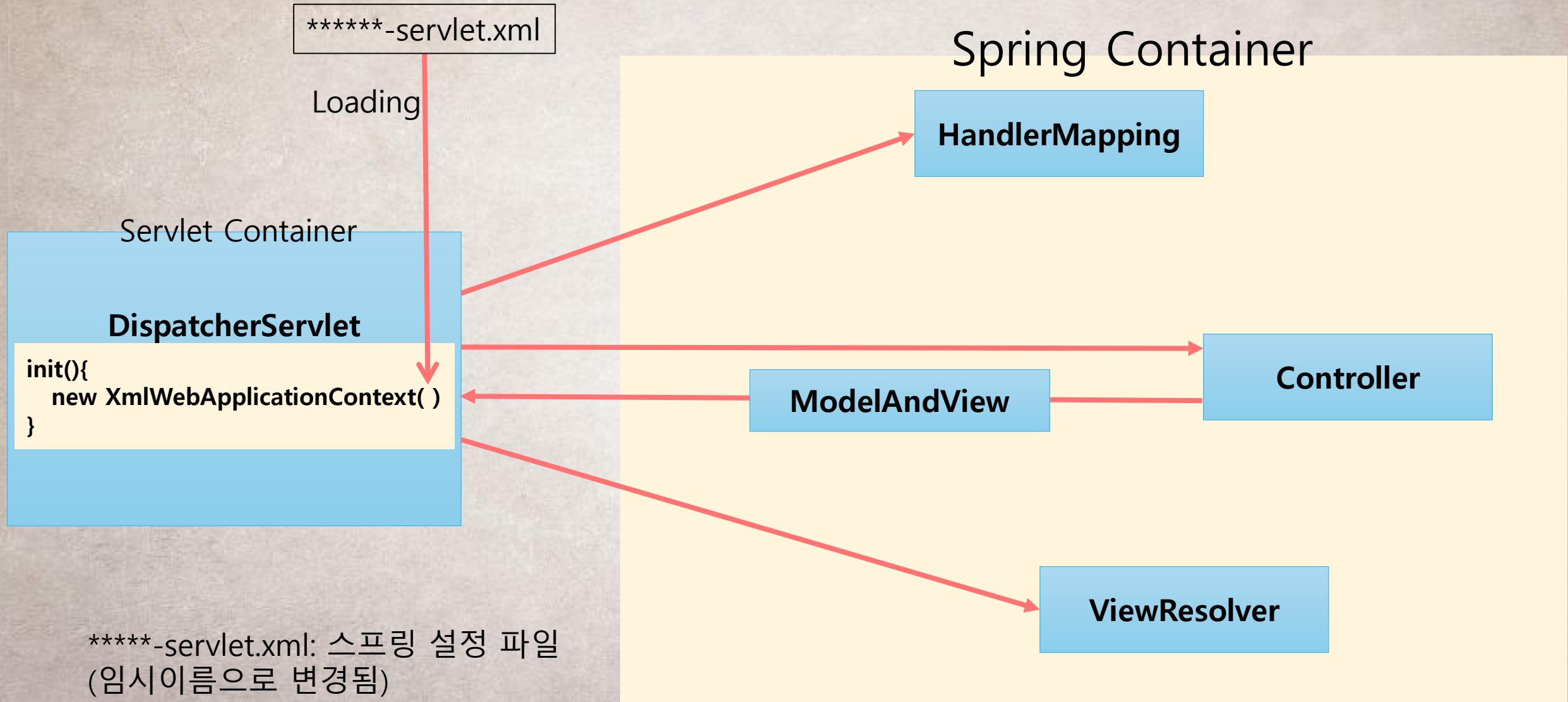
- SpringMVC의 구성요소 중에서 DispatcherServlet 클래스가 유일한 서블릿이고 web.xml 파일에서 등록된 **DispatcherServlet**만 생성
- DispatcherServlet 객체 혼자서는 클라이언트의 요청을 처리할 수 없고, 반드시 **HandlerMapping, Controller, ViewResolver** 객체들과 상호작용해야 한다. 이들을 사용하기 위해 DispatcherServlet은 스프링 컨테이너를 구동하는 것임.
- 사용자가 직접 DispatcherServlet 생성할 경우에는 init()에서 직접 생성했지만 스프링에서는 스프링컨테이너를 통해 객체를 생성하는 것이 다름.

➔ 사용자가 직접 DispatcherServlet을 생성한 경우

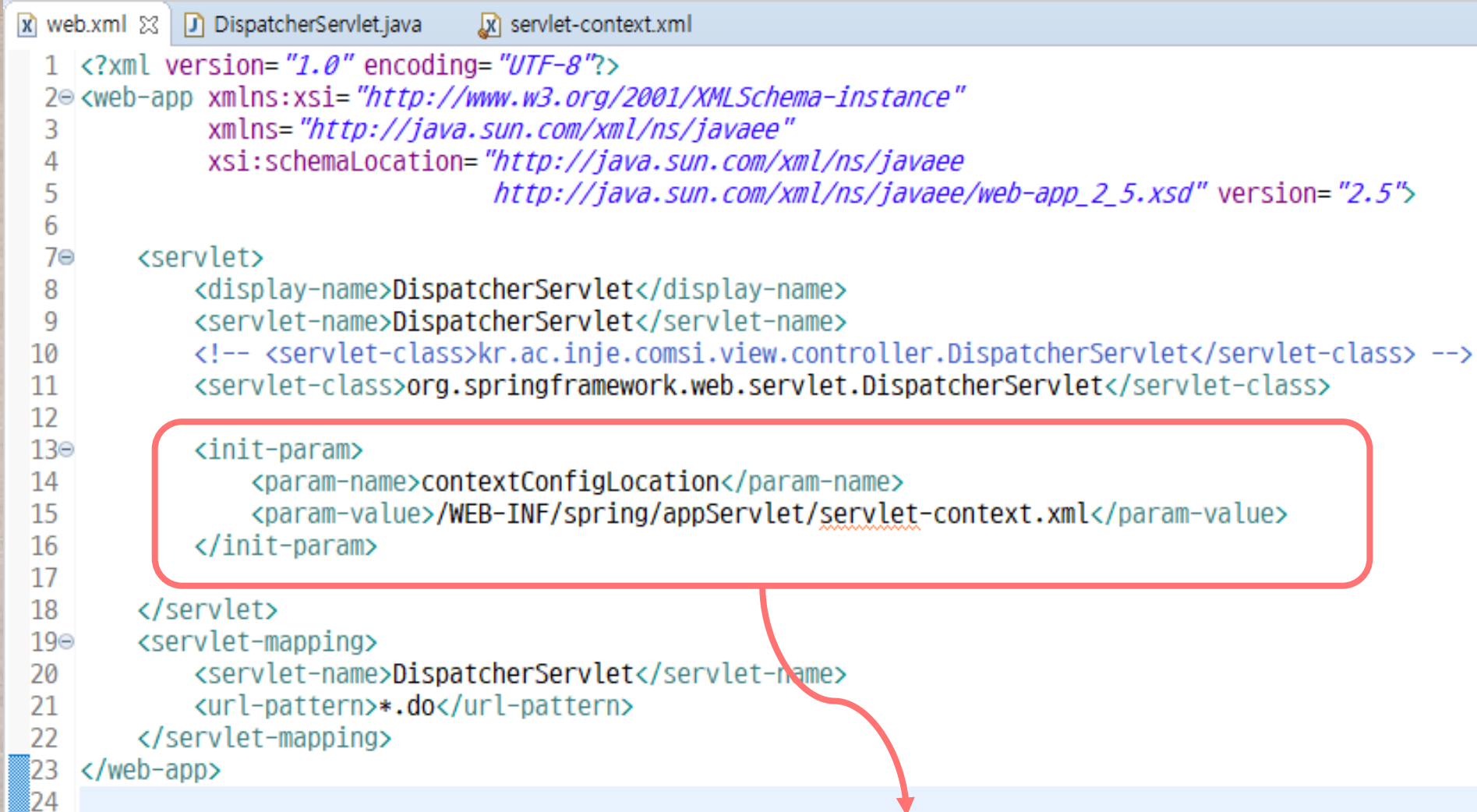
```
12 public class DispatcherServlet extends HttpServlet {
13     private static final long serialVersionUID = 1L;
14     private HandlerMapping handlerMapping;
15     private ViewResolver viewResolver;
16
17     @Override
18     public void init() throws ServletException {
19         handlerMapping = new HandlerMapping();
20         viewResolver = new ViewResolver();
21         viewResolver.setPrefix("./");
22         viewResolver.setSuffix(".jsp");
23     }
24
```

직접 생성

스프링 컨테이너(XmlWebApplicatonContext)의 구동 과정



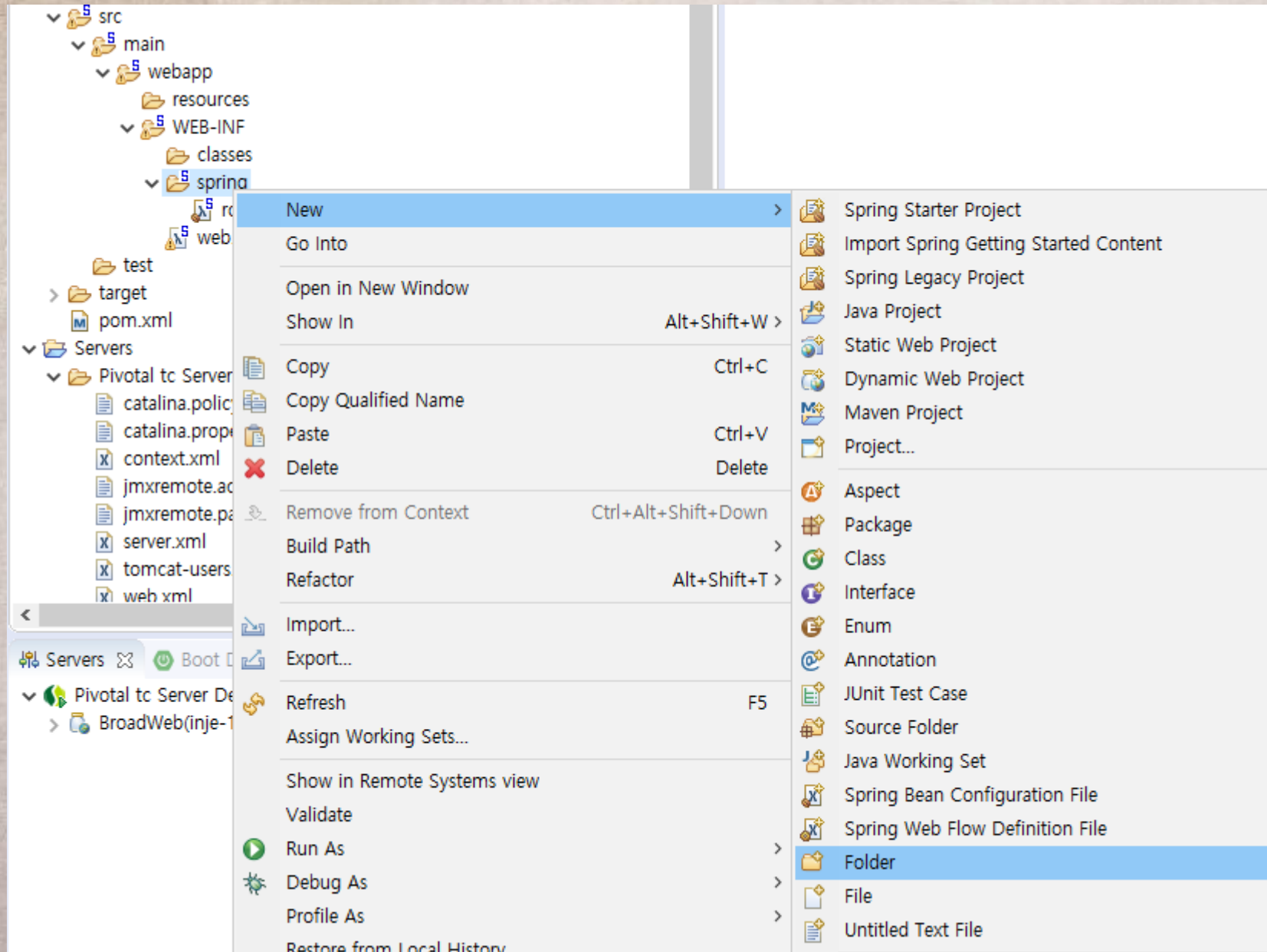
5.3 스프링 설정 파일 등록-web.xml 수정



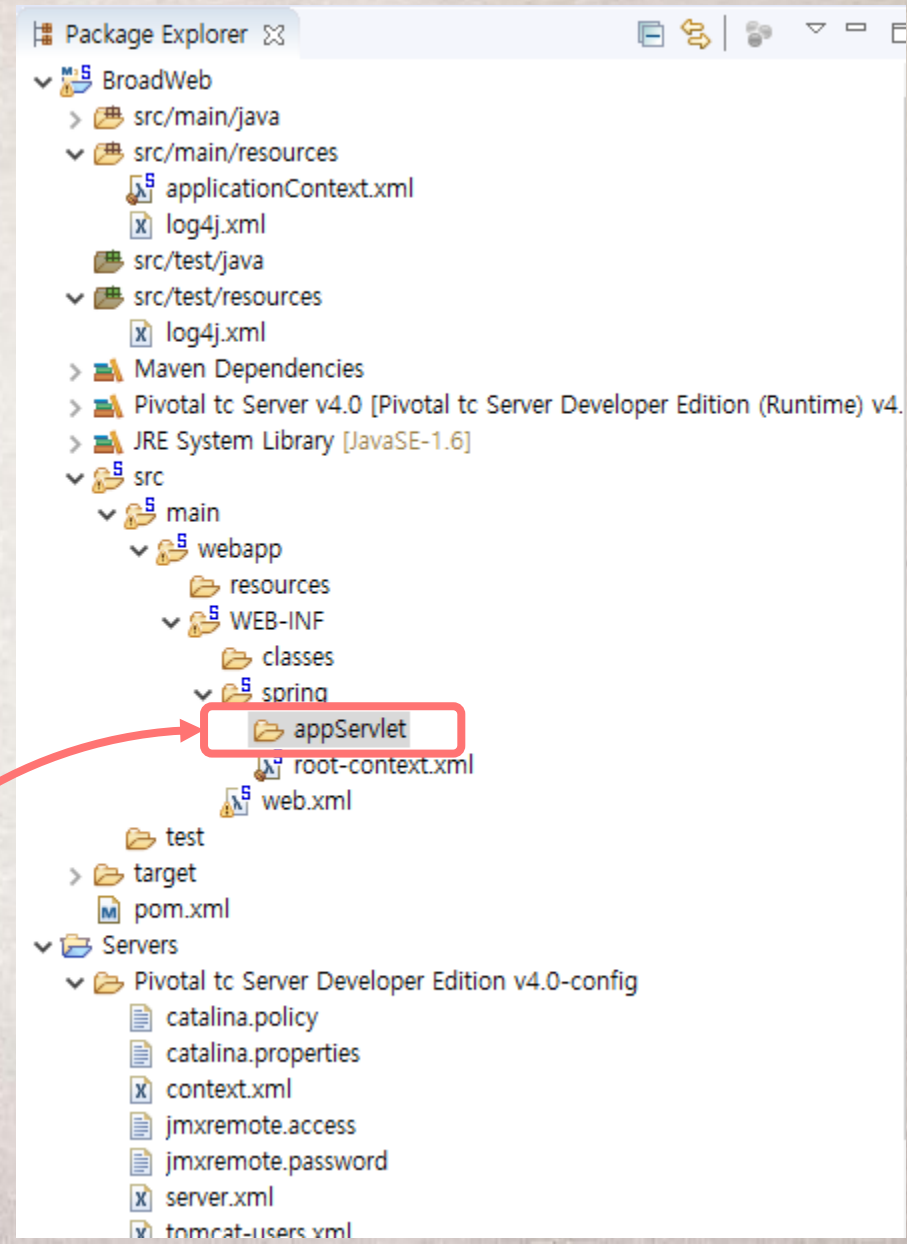
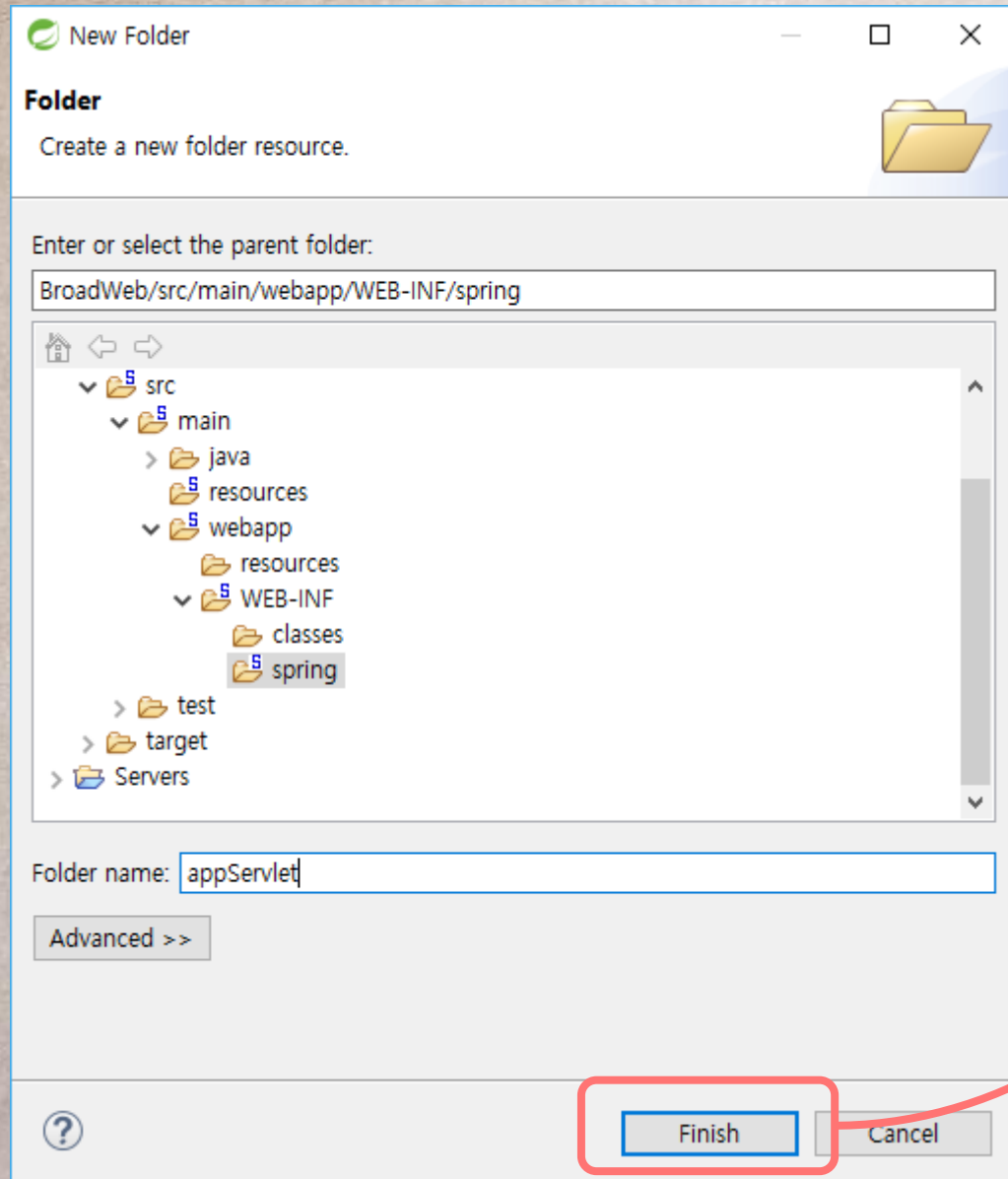
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://java.sun.com/xml/ns/javaee"
4         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
5                             http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
6
7     <servlet>
8         <display-name>DispatcherServlet</display-name>
9         <servlet-name>DispatcherServlet</servlet-name>
10        <!-- <servlet-class>kr.ac.inje.comsi.view.controller.DispatcherServlet</servlet-class> -->
11        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
12
13        <init-param>
14            <param-name>contextConfigLocation</param-name>
15            <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
16        </init-param>
17
18    </servlet>
19    <servlet-mapping>
20        <servlet-name>DispatcherServlet</servlet-name>
21        <url-pattern>*.do</url-pattern>
22    </servlet-mapping>
23 </web-app>
24
```

스프링 컨테이너가 DispatcherServlet 객체를 생성한 후 init() 메소드를 호출하여 "**contextConfigLocation**"이라는 파라미터로 설정한 정보를 추출하여 스프링 컨테이너를 구동할 때 사용한다.

/WEB-INF/spring/appServlet/servlet-context.xml



/WEB-INF/spring/appServlet/servlet-context.xml

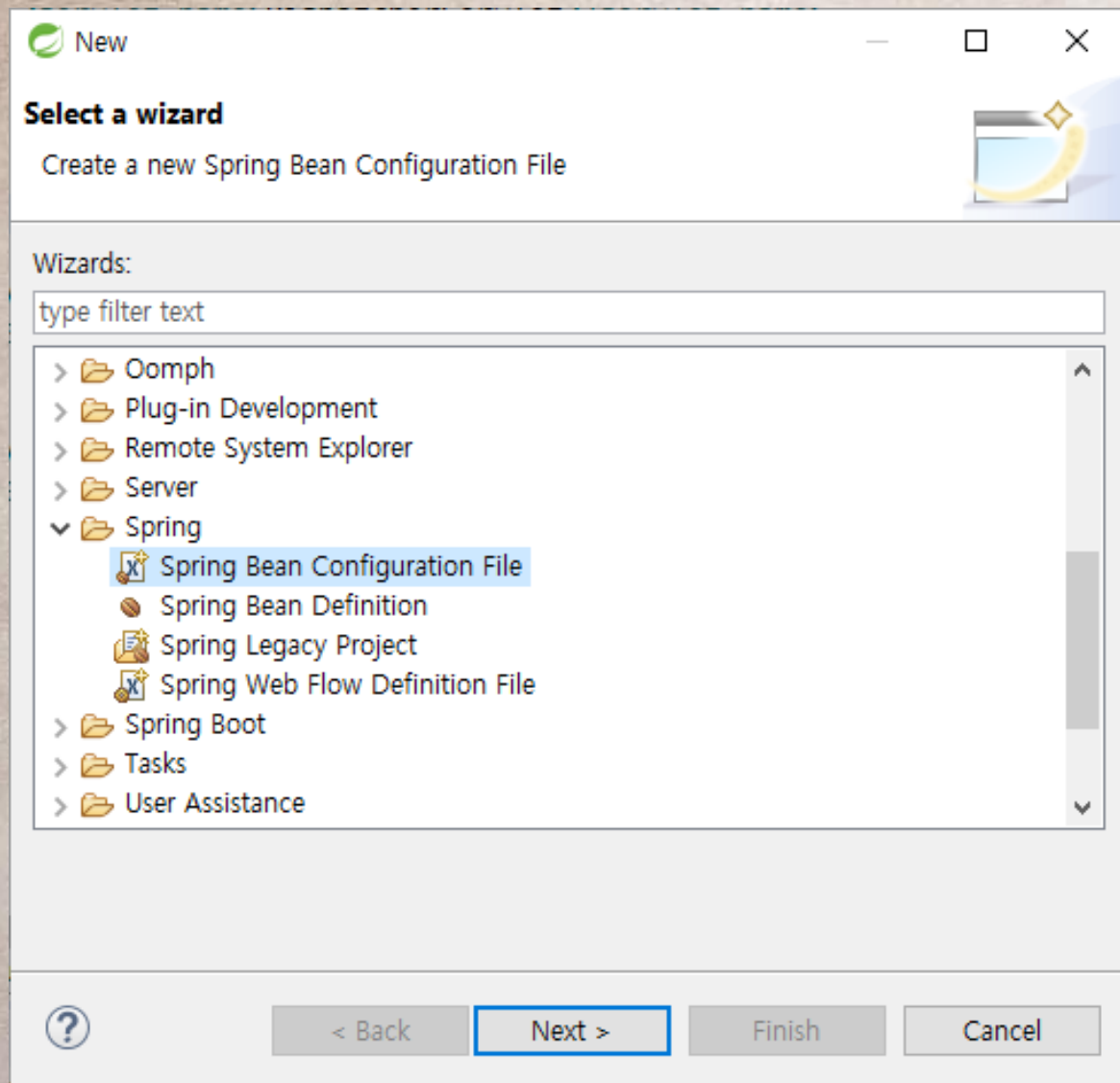


/WEB-INF/spring/appServlet/servlet-context.xml

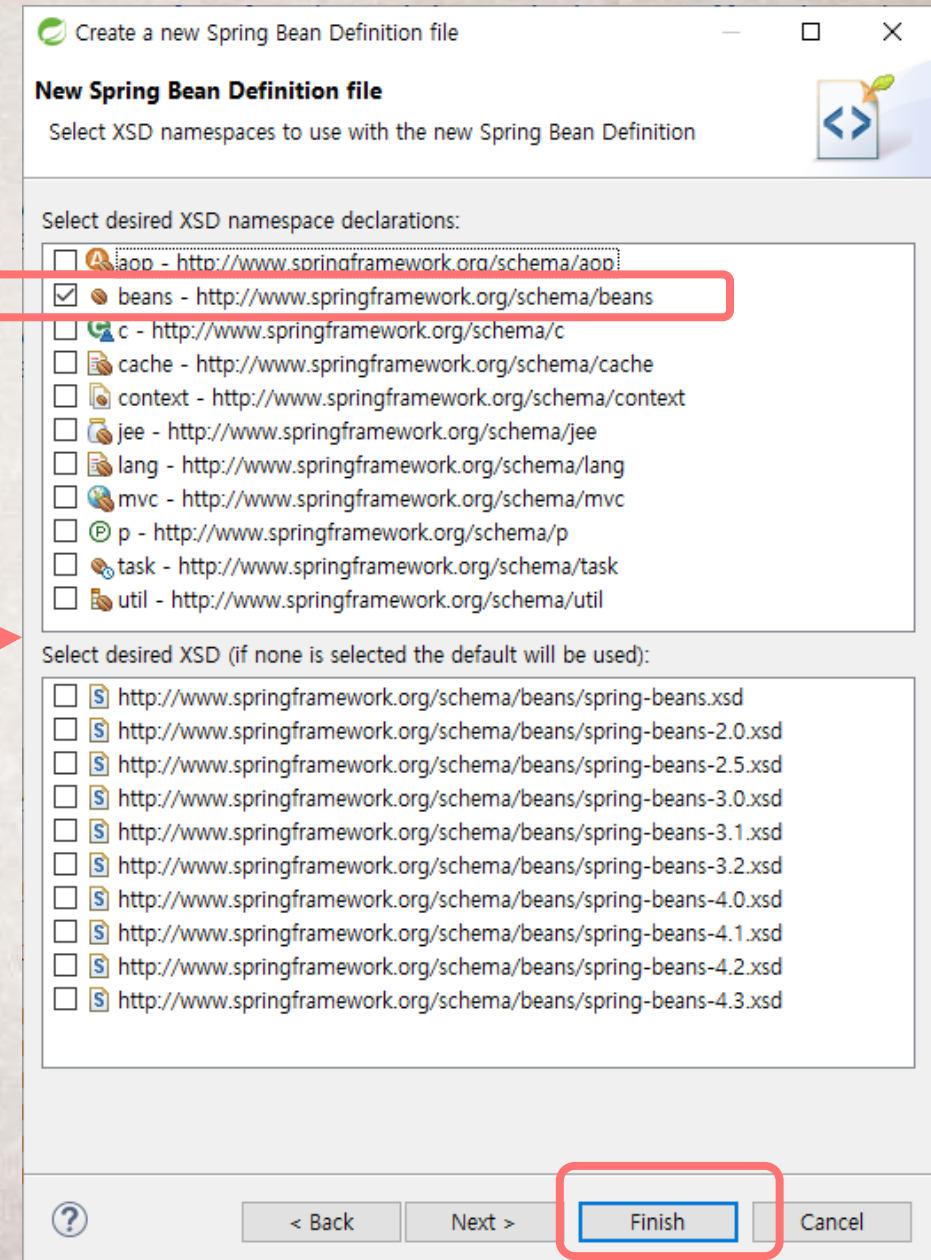
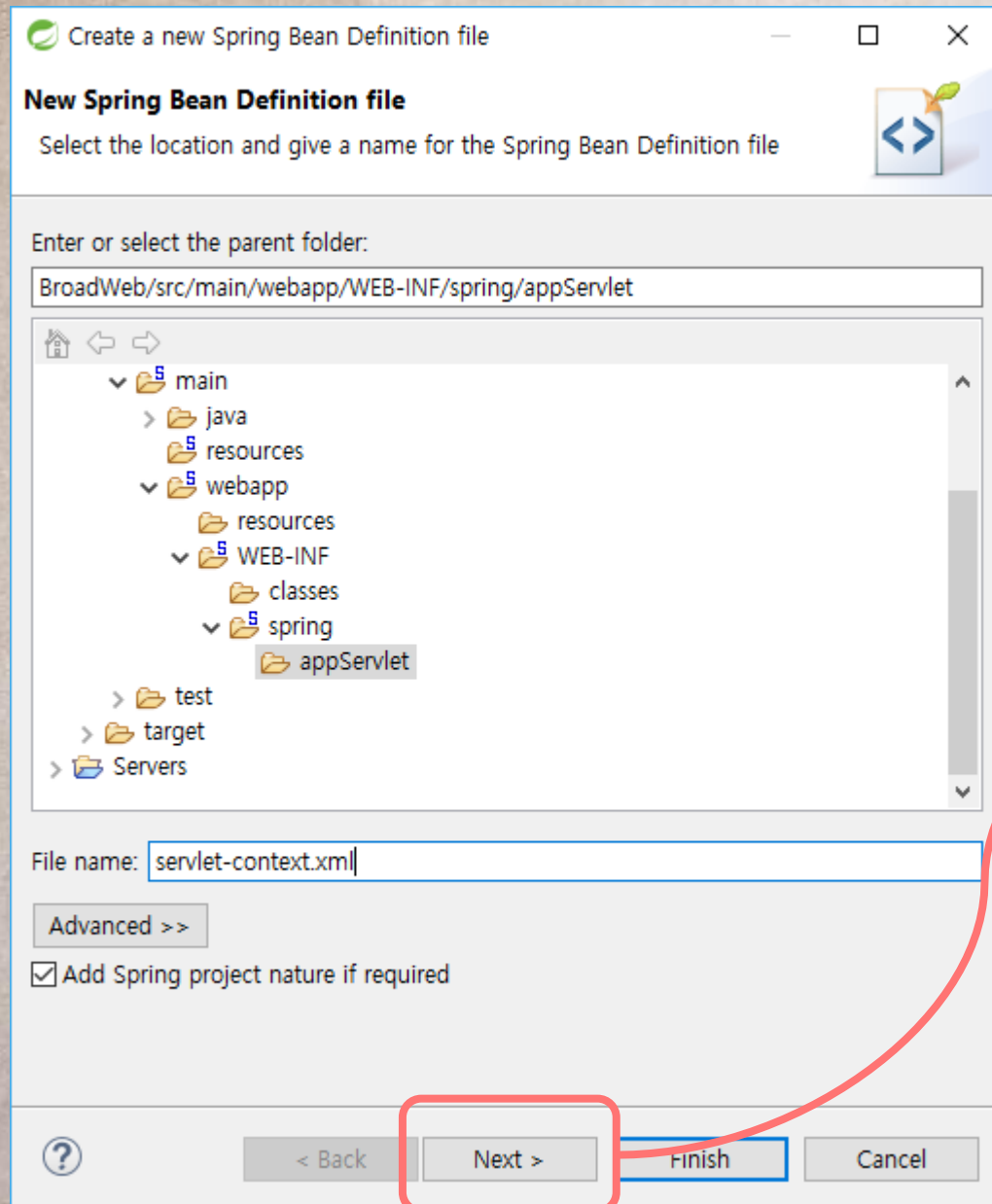
The screenshot shows an IDE interface with a project structure on the left and a code editor on the right. The project structure includes a 'src' directory with 'main' and 'test' subdirectories. The 'main' directory contains 'webapp', 'resources', 'WEB-INF', 'classes', and 'spring'. The 'spring' directory contains 'appServlet'. The 'WEB-INF' directory contains 'classes' and 'spring'. The 'spring' directory contains 'appServlet'. The 'appServlet' directory contains 'root' and 'web.xml'. The 'root' directory contains 'context.xml'. The 'web.xml' file is selected, and a context menu is open over it. The context menu includes options like 'New', 'Open in New Window', 'Show In', 'Copy', 'Copy Qualified Name', 'Paste', 'Delete', 'Remove from Context', 'Build Path', 'Refactor', 'Import...', 'Export...', 'Refresh', 'Assign Working Sets...', 'Show in Remote Systems view', 'Validate', 'Run As', and 'Debug As'. The 'New' option is expanded, showing a list of new elements that can be created, including 'Spring Starter Project', 'Import Spring Getting Started Content', 'Spring Legacy Project', 'Java Project', 'Static Web Project', 'Dynamic Web Project', 'Maven Project', 'Project...', 'Aspect', 'Package', 'Class', 'Interface', 'Enum', 'Annotation', 'JUnit Test Case', 'Source Folder', 'Java Working Set', 'Spring Bean Configuration File' (which is highlighted), 'Spring Web Flow Definition File', and 'Folder'. The code editor on the right shows the content of the 'context.xml' file, which includes XML tags for creating a Spring container and processing application requests.

```
12
13
14 <!-- Creates the Spring Container shared with
15 <listener>
16     <listener-class>org.springframework.web
17 </listener>
18
19 <!-- Processes application requests -->
20 <servlet>
```

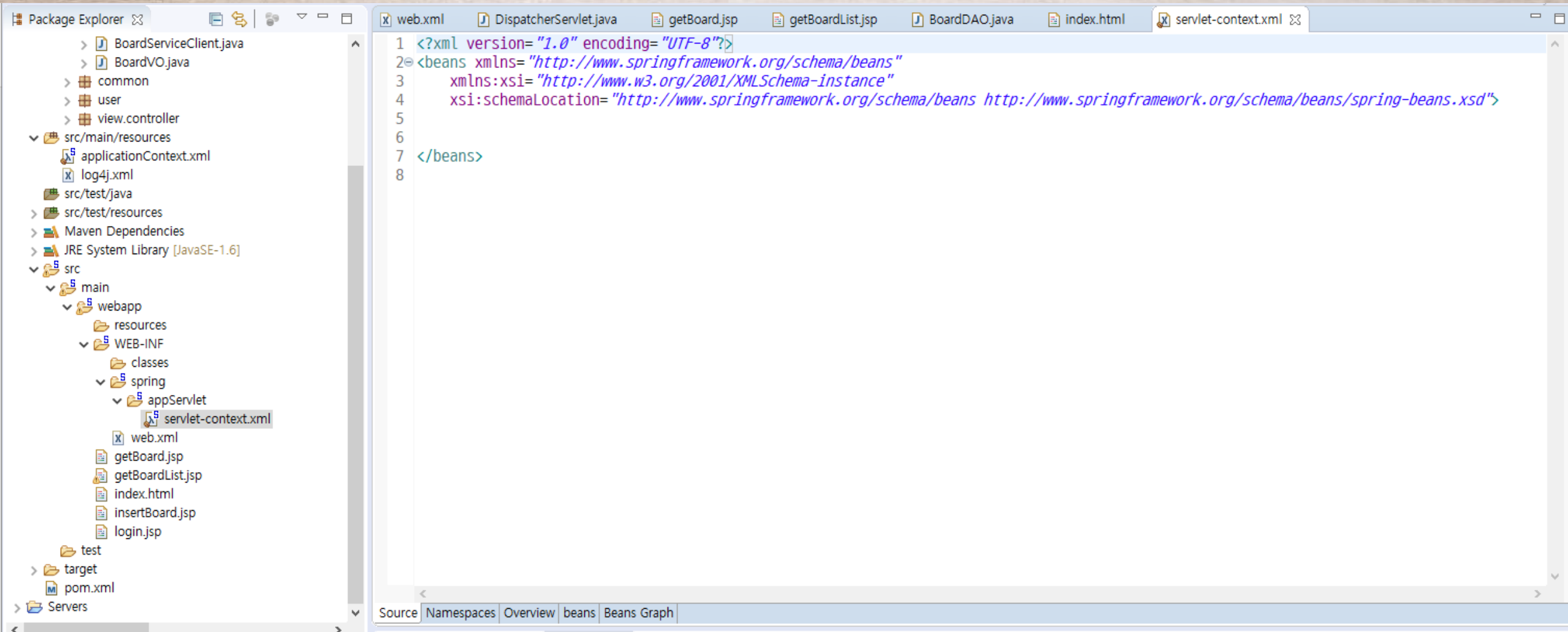
폴더 메뉴에서 Bean configuration file 메뉴가 안 뜰때



/WEB-INF/spring/appServlet/servlet-context.xml



/WEB-INF/spring/appServlet/servlet-context.xml



The screenshot displays an IDE interface with a Package Explorer on the left and a code editor on the right. The Package Explorer shows a project structure with folders like 'src/main/resources', 'src/test/java', 'Maven Dependencies', 'JRE System Library [JavaSE-1.6]', 'src', 'main', 'webapp', 'resources', 'WEB-INF', 'classes', 'spring', 'appServlet', and 'test'. The 'servlet-context.xml' file is selected under 'appServlet'.

The code editor shows the content of 'servlet-context.xml' with the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
5
6
7 </beans>
8
```

The bottom of the IDE shows a tabbed interface with 'Source', 'Namespaces', 'Overview', 'beans', and 'Beans Graph' tabs. The 'Source' tab is currently active.

/login.do 호출

web.xml DispatcherServlet.java getBoard.jsp getBoardList.jsp BoardDAO.java index.html servlet-context.xml HTTP Status 404 – Not Found

http://localhost:8080/Board/login.do

HTTP Status 404 – Not Found

Type Status Report

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Pivotal tc Runtime 9.0.10.A.RELEASE

@ Javadoc Declaration Console Progress

Pivotal tc Server Developer Edition v4.0 [Pivotal tc Server] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (2018. 10. 12. 오전 8:23:18)

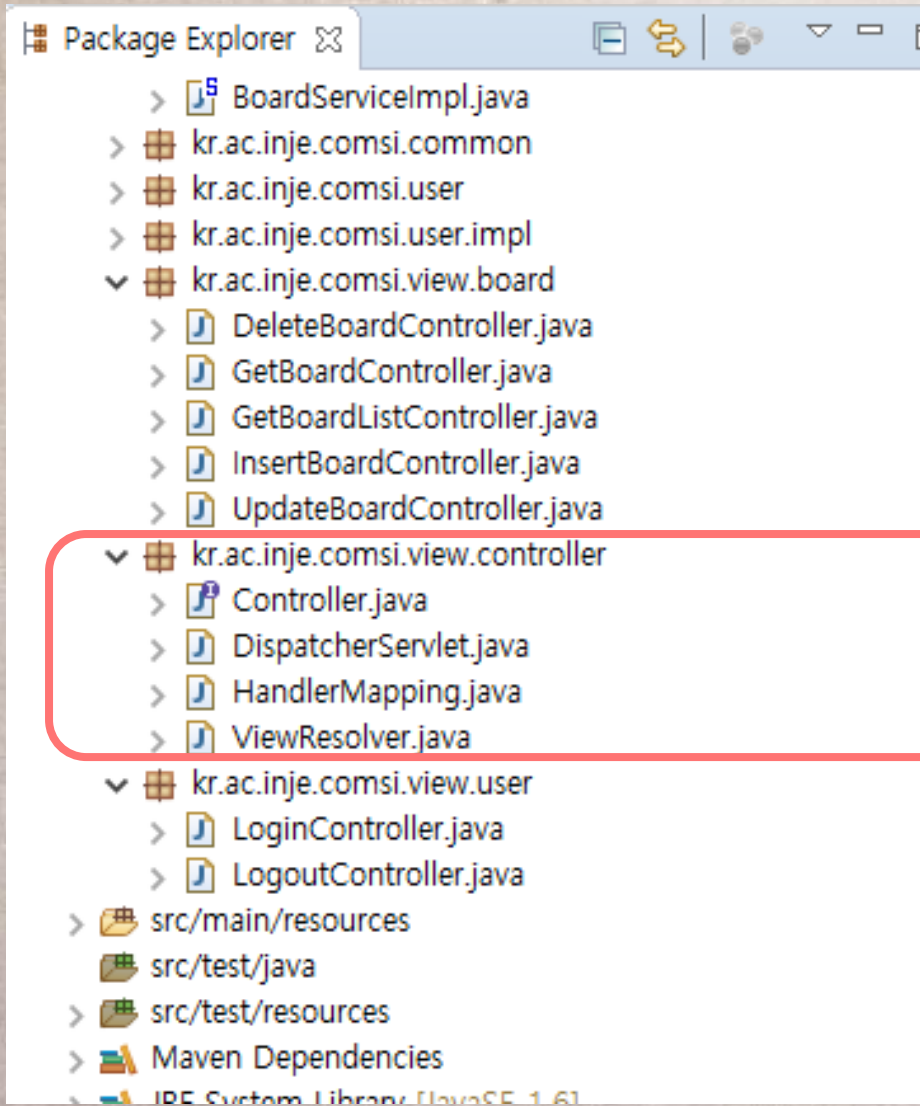
10월 12, 2018 8:23:20 오전 org.apache.catalina.startup.Catalina load
정보: Initialization processed in 1402 ms
10월 12, 2018 8:23:21 오전 org.apache.catalina.startup.Catalina start
정보: Server startup in 1601 ms
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'DispatcherServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'DispatcherServlet-servlet': startup date
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/spring/appServlet/servlet-context.xml]
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'DispatcherServlet': initialization completed in 403 ms
WARN : org.springframework.web.servlet.PageNotFound - No mapping found for HTTP request with URI [/Board/login.do] in DispatcherServlet with name 'DispatcherServlet'

➔ login.do를 처리할 대상이 Controller가 Mapping되어 있지않다.

6~7 Spring MVC 적용

6.1 Spring MVC 적용 준비

- kr.ac.inje.comsi.view.controller 패키지를 삭제



삭제한다(스프링 컨테이너에서 처리할 것임).

(1) LoginController 구현 - 수정

- 유저가 생성한 Controller 인터페이스를 스프링 Controller 인터페이스로 변경
- Controller 인터페이스의 `handleRequest`함수의 리턴타입을 String에서 **ModelAndView** 객체로 변환

```
package org.springframework.web.servlet.mvc; ↵
↵
import javax.servlet.http.HttpServletRequest; ↵
import javax.servlet.http.HttpServletResponse; ↵
↵
import org.springframework.web.servlet.ModelAndView; ↵
↵
public interface Controller { ↵
    ModelAndView handleRequest(HttpServletRequest request, ↵
                               HttpServletResponse response) throws Exception; ↵
↵
} ↵
```


6.2 로그인 기능 구현 – (1)LoginController 구현-수정

```
9 import org.springframework.web.servlet.ModelAndView;
10 //import kr.ac.inje.comsi.view.controller.Controller;
11 import org.springframework.web.servlet.mvc.Controller;
12
13 public class LoginController implements Controller {
14
15     @Override
16     public ModelAndView handleRequest(HttpServletRequest request, HttpServletResponse response) {
17         System.out.println("로그인 처리");
18
19         // 1. 사용자 입력 정보 추출
20         String id = request.getParameter("id");
21         String password = request.getParameter("password");
22
23         // 2. DB 연동 처리
24         UserVO vo = new UserVO();
25         vo.setId(id);
26         vo.setPassword(password);
27
28         UserDAO userDAO = new UserDAO();
29         UserVO user = userDAO.getUser(vo);
30
31         // 3. 화면 이동
32         ModelAndView mav = new ModelAndView();
33         if (user != null) {
34             mav.setViewName("getBoardList.do");
35         } else {
36             mav.setViewName("login.jsp");
37         }
38         return mav;
39     }
40 }
41
```

(2) HandlerMapping 등록 → servlet-context.xml

```
<bean class = "org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name = "mappings">
        <props>
            <prop key = "/welcome.htm">welcomeController</prop>
            <prop key = "/helloWorld.htm">helloController</prop>
        </props>
    </property>
</bean>
```

```
<bean id="welcomeController" class="kr.ac.inje.com.WelcomeController"/>
<bean id="helloController" class="kr.ac.inje.com.HelloController"/>
```

- SimpleUrlHandlerMapping 설정

(2) HandlerMapping 등록 → servlet-context.xml

```
web.xml | servlet-context.xml | LoginController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:mvc="http://www.springframework.org/schema/mvc"
6       xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
7                           http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
8                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
9
10
11 <!-- HandlerMapping 등록 -->
12 <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
13   <property name="mappings">
14     <props>
15       <prop key="/login.do">login</prop>
16     </props>
17   </property>
18 </bean>
19 <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
20 </beans>
21
22
```

추가

- 상자의 내용은 앞에서 직접 구현한 HandlerMapping과 같다.

6.3 글 목록 검색 기능 구현-(1)GetBoardListController 구현

```
16 public class GetBoardListController implements Controller {
17
18     @Override
19     public ModelAndView handleRequest(HttpServletRequest request,
20                                     HttpServletResponse response) {
21         System.out.println("글 목록 검색 처리");
22
23         // 1. 사용자 입력 정보 추출(검색 기능은 나중에~~)
24         // 2. DB 연동 처리
25         BoardVO vo = new BoardVO();
26         BoardDAO boardDAO = new BoardDAO();
27         List<BoardVO> boardList = boardDAO.getBoardList(vo);
28
29         // 3. 검색 결과를 ModelAndView에 저장하고 목록 화면을 리턴
30         ModelAndView mav = new ModelAndView();
31         mav.addObject("boardList", boardList);
32         mav.setViewName("getBoardList.jsp");
33         return mav;
34     }
35 }
```

(2) HandlerMapping 등록 → servlet-context.xml

```
web.xml  servlet-context.xml  LoginController.java  GetBoardListController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:mvc="http://www.springframework.org/schema/mvc"
6     xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
7     http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
8     http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
9
10
11     <!-- HandlerMapping 등록 -->
12     <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
13         <property name="mappings">
14             <props>
15                 <prop key="/login.do">login</prop>
16                 <prop key="/getBoardList.do">getBoardList</prop>
17             </props>
18         </property>
19     </bean>
20     <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
21     <bean id="getBoardList" class="kr.ac.inje.comsi.view.board.GetBoardListController"></bean>
22 </beans>
23
24
```

6.4 글 상세 조회 기능-(1)GetBoardController 구현

```
web.xml  servlet-context.xml  GetBoardController.java  ⌕
1 package kr.ac.inje.comsi.view.board;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6 import org.springframework.web.servlet.ModelAndView;
7 import org.springframework.web.servlet.mvc.Controller;
8
9 import kr.ac.inje.comsi.board.BoardVO;
10 import kr.ac.inje.comsi.board.impl.BoardDAO;
11 //import kr.ac.inje.comsi.view.controller.Controller;
12
13 public class GetBoardController implements Controller {
14
15     @Override
16     public ModelAndView handleRequest(HttpServletRequest request,
17                                     HttpServletResponse response) {
18         System.out.println("글 상세 조회 처리");
19
20         // 1. 검색할 게시글 번호 추출
21         String seq = request.getParameter("seq");
22
23         // 2. DB 연동 처리
24         BoardVO vo = new BoardVO();
25         vo.setSeq(Integer.parseInt(seq));
26
27         BoardDAO boardDAO = new BoardDAO();
28         BoardVO board = boardDAO.getBoard(vo);
29
30         // 3. 검색 결과를 ModelAndView에 저장하고 목록 화면을 리턴
31         ModelAndView mav = new ModelAndView();
32         mav.addObject("board", board);
33         mav.setViewName("getBoard.jsp");
34         return mav; //
35     }
36 }
```


(2) HandlerMapping 등록 → servlet-context.xml

```
web.xml | servlet-context.xml | GetBoardController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:mvc="http://www.springframework.org/schema/mvc"
6       xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
7                           http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
8                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
9
10
11     <!-- HandlerMapping 등록 -->
12     <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
13         <property name="mappings">
14             <props>
15                 <prop key="/login.do">login</prop>
16                 <prop key="/getBoardList.do">getBoardList</prop>
17                 <prop key="/getBoard.do">getBoard</prop>
18             </props>
19         </property>
20     </bean>
21     <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
22     <bean id="getBoardList" class="kr.ac.inje.comsi.view.board.GetBoardListController"></bean>
23     <bean id="getBoard" class="kr.ac.inje.comsi.view.board.GetBoardController"></bean>
24 </beans>
25
26
```

6.5 글 등록 기능-(1)InsertBoardController 구현

```
web.xml  servlet-context.xml  GetBoardController.java  InsertBoardController.java  ✕
1 package kr.ac.inje.comsi.view.board;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6 import org.springframework.web.servlet.ModelAndView;
7 import org.springframework.web.servlet.mvc.Controller;
8
9 import kr.ac.inje.comsi.board.BoardVO;
10 import kr.ac.inje.comsi.board.impl.BoardDAO;
11 //import kr.ac.inje.comsi.view.controller.Controller;
12
13 public class InsertBoardController implements Controller {
14
15     @Override
16     public ModelAndView handleRequest(HttpServletRequest request,
17                                     HttpServletResponse response) {
18         System.out.println("글 등록 처리");
19
20         // 1. 사용자 입력한 정보 추출
21         String title = request.getParameter("title");
22         String content = request.getParameter("content");
23         String writer = request.getParameter("writer");
24
25         // 2. DB 연동 처리
26         BoardVO vo = new BoardVO();
27         vo.setTitle(title);
28         vo.setContent(content);
29         vo.setWriter(writer);
30
31         BoardDAO boardDAO = new BoardDAO();
32         boardDAO.insertBoard(vo);
33
34         // 3. 화면을 리턴
35         ModelAndView mav = new ModelAndView();
36         mav.setViewName("getBoardList.do");
37         return mav; //
38     }
39 }
```

(2) HandlerMapping 등록 → servlet-context.xml

```
web.xml  servlet-context.xml  GetBoardController.java  InsertBoardController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:mvc="http://www.springframework.org/schema/mvc"
6     xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
7         http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
8         http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
9
10
11 <!-- HandlerMapping 등록 -->
12 <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
13     <property name="mappings">
14         <props>
15             <prop key="/login.do">login</prop>
16             <prop key="/getBoardList.do">getBoardList</prop>
17             <prop key="/getBoard.do">getBoard</prop>
18             <prop key="/insertBoard.do">insertBoard</prop>
19         </props>
20     </property>
21 </bean>
22 <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
23 <bean id="getBoardList" class="kr.ac.inje.comsi.view.board.GetBoardListController"></bean>
24 <bean id="getBoard" class="kr.ac.inje.comsi.view.board.GetBoardController"></bean>
25 <bean id="insertBoard" class="kr.ac.inje.comsi.view.board.InsertBoardController"></bean>
26 </beans>
27
```


6.6 글 수정 기능-(1)UpdateBoardController 구현

```
servlet-context.xml DeleteBoardController.java UpdateBoardController.java
1 package kr.ac.inje.comsi.view.board;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6 import org.springframework.web.servlet.ModelAndView;
7 import org.springframework.web.servlet.mvc.Controller;
8
9 import kr.ac.inje.comsi.board.BoardVO;
10 import kr.ac.inje.comsi.board.impl.BoardDAO;
11 //import kr.ac.inje.comsi.view.controller.Controller;
12
13 public class UpdateBoardController implements Controller {
14
15     @Override
16     public ModelAndView handleRequest(HttpServletRequest request,
17                                     HttpServletResponse response) {
18         System.out.println("글 수정 처리");
19
20         // 1. 사용자 입력한 정보 추출
21         String title = request.getParameter("title");
22         String content = request.getParameter("content");
23         String seq = request.getParameter("seq");
24
25         // 2. DB 연동 처리
26         BoardVO vo = new BoardVO();
27         vo.setTitle(title);
28         vo.setContent(content);
29         vo.setSeq(Integer.parseInt(seq));
30
31         BoardDAO boardDAO = new BoardDAO();
32         boardDAO.updateBoard(vo);
33
34         // 3. 화면을 리턴
35         ModelAndView mav = new ModelAndView();
36         mav.setViewName("getBoardList.do");
37         return mav;
38     }
39 }
```

6.7 글 삭제 기능-(1)DeleteBoardController 구현

```
servlet-context.xml DeleteBoardController.java UpdateBoardController.java
1 package kr.ac.inje.comsi.view.board;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6 import org.springframework.web.servlet.ModelAndView;
7 import org.springframework.web.servlet.mvc.Controller;
8
9 import kr.ac.inje.comsi.board.BoardVO;
10 import kr.ac.inje.comsi.board.impl.BoardDAO;
11 //import kr.ac.inje.comsi.view.controller.Controller;
12
13 public class DeleteBoardController implements Controller {
14
15     @Override
16     public ModelAndView handleRequest(HttpServletRequest request,
17                                     HttpServletResponse response) {
18         System.out.println("글 삭제 처리");
19
20         // 1. 삭제할 게시글 번호 추출
21         String seq = request.getParameter("seq");
22
23         // 2. DB 연동 처리
24         BoardVO vo = new BoardVO();
25         vo.setSeq(Integer.parseInt(seq));
26
27         BoardDAO boardDAO = new BoardDAO();
28         boardDAO.deleteBoard(vo);
29
30         // 3. 삭제하고 목록 화면을 리턴
31         ModelAndView mav = new ModelAndView();
32         mav.setViewName("getBoardList.do");
33         return mav;
34     }
35 }
```

Handlermapping 등록

```
servlet-context.xml DeleteBoardController.java UpdateBoardController.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:mvc="http://www.springframework.org/schema/mvc"
6       xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
7                           http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
8                           http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
9
10
11 <!-- HandlerMapping 등록 -->
12 <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
13     <property name="mappings">
14         <props>
15             <prop key="/login.do">login</prop>
16             <prop key="/getBoardList.do">getBoardList</prop>
17             <prop key="/getBoard.do">getBoard</prop>
18             <prop key="/insertBoard.do">insertBoard</prop>
19             <prop key="/updateBoard.do">updateBoard</prop>
20             <prop key="/deleteBoard.do">deleteBoard</prop>
21         </props>
22     </property>
23 </bean>
24 <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
25 <bean id="getBoardList" class="kr.ac.inje.comsi.view.board.GetBoardListController"></bean>
26 <bean id="getBoard" class="kr.ac.inje.comsi.view.board.GetBoardController"></bean>
27 <bean id="insertBoard" class="kr.ac.inje.comsi.view.board.InsertBoardController"></bean>
28 <bean id="updateBoard" class="kr.ac.inje.comsi.view.board.UpdateBoardController"></bean>
29 <bean id="deleteBoard" class="kr.ac.inje.comsi.view.board.DeleteBoardController"></bean>
30 </beans>
31
```


6.8 로그아웃 기능 구현하기-LogoutController

```
servlet-context.xml DeleteBoardController.java UpdateBoardController.java LogoutController.java
1 package kr.ac.inje.comsi.view.user;
2
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5 import javax.servlet.http.HttpSession;
6
7 import org.springframework.web.servlet.ModelAndView;
8 import org.springframework.web.servlet.mvc.Controller;
9
10 //import kr.ac.inje.comsi.view.controller.Controller;
11
12 public class LogoutController implements Controller {
13
14     @Override
15     public ModelAndView handleRequest(HttpServletRequest request,
16                                     HttpServletResponse response) {
17         System.out.println("로그아웃 처리");
18
19         // 1. 브라우저와 연결된 세션 객체를 강제 종료
20         HttpSession session = request.getSession();
21         session.invalidate();
22         // 2. 세션 종료 후, 메인 화면으로 이동
23         ModelAndView mav = new ModelAndView();
24         mav.setViewName("login.jsp");
25         return mav;
26     }
27 }
```

Handlermapping 등록

```
10  <!-- HandlerMapping 등록 -->
11  <bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
12      <property name="mappings">
13          <props>
14              <prop key="/login.do">login</prop>
15              <prop key="/getBoardList.do">getBoardList</prop>
16              <prop key="/getBoard.do">getBoard</prop>
17              <prop key="/insertBoard.do">insertBoard</prop>
18              <prop key="/updateBoard.do">updateBoard</prop>
19              <prop key="/deleteBoard.do">deleteBoard</prop>
20              <prop key="/logout.do">logoutBoard</prop>
21          </props>
22      </property>
23  </bean>
24  <bean id="login" class="kr.ac.inje.comsi.view.user.LoginController"></bean>
25  <bean id="getBoardList" class="kr.ac.inje.comsi.view.board.GetBoardListController"></bean>
26  <bean id="getBoard" class="kr.ac.inje.comsi.view.board.GetBoardController"></bean>
27  <bean id="insertBoard" class="kr.ac.inje.comsi.view.board.InsertBoardController"></bean>
28  <bean id="updateBoard" class="kr.ac.inje.comsi.view.board.UpdateBoardController"></bean>
29  <bean id="deleteBoard" class="kr.ac.inje.comsi.view.board.DeleteBoardController"></bean>
30  <bean id="logoutBoard" class="kr.ac.inje.comsi.view.user.LogoutController"></bean>
31
```


View 페이지(JSP) 수정

EL과 JSTL

- EL(Expression Language)은 JSP 2.0에서 추가된 스크립트 언어로 기존 표현식을 대체하는 표현언어
- `<%=session.getAttribute("username")%>` → `${username}`
- JSTL(JSP Standard Tag Library)는 scriptlet에서 **if, for, switch**등과 같은 자바 코드를 태그 형태로 사용할 수 있도록 지원
- 참고 사이트: <http://www.tutorialspoint.com/jsp/>

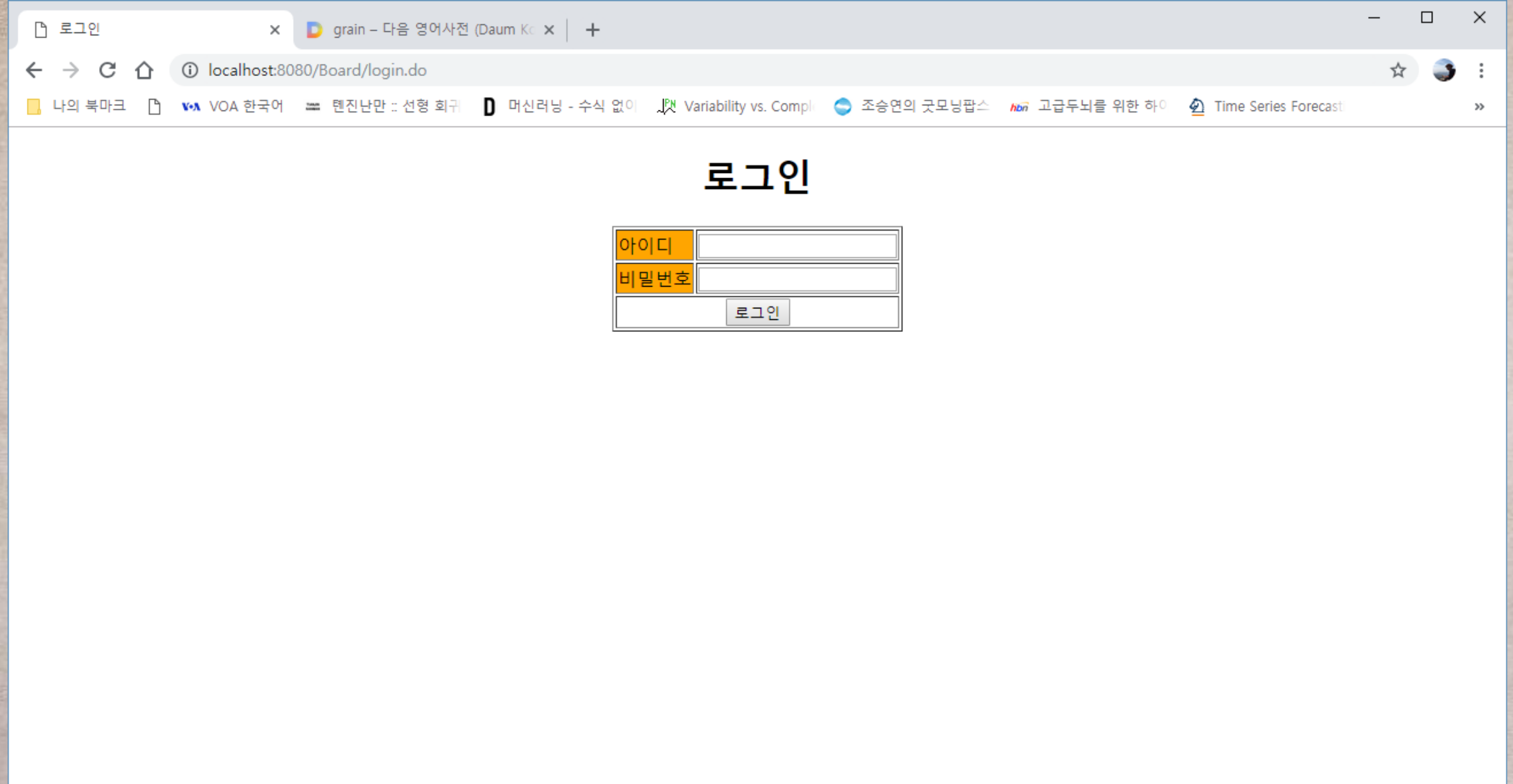
14-
15
16-
17-
18
19-
20
21
22-
23
24
25
26-
27
28-
29
30
31-
32
33
34
35-
36
37
38
39-
40-
41
42
43
44
45
46
47
48

getBoardList.jsp(EL 표현언어로 수정)

```
servlet-context.xml DeleteBoardController.java getBoardList.jsp getBoard.jsp login.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ taglib uri="http://java.sun.com/jstl/core_rt" prefix="c" %>
4 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
5 <html>
6 <head>
7   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8   <title>글 목록</title>
9 </head>
```

```
32 <table border="1" width="700">
33 <tr>
34   <th bgcolor="orange" width="100">번호</th>
35   <th bgcolor="orange" width="200">제목</th>
36   <th bgcolor="orange" width="150">작성자</th>
37   <th bgcolor="orange" width="150">등록일</th>
38   <th bgcolor="orange" width="100">조회수</th>
39 </tr>
40 <c:forEach items="${boardList}" var="board">
41 <tr>
42   <td align="center">${board.seq}</td>
43   <td align="left"><a href="getBoard.do?seq=${board.seq}">${board.title}</a></td>
44   <td align="center">${board.writer}</td>
45   <td align="center">${board.regDate}</td>
46   <td align="center">${board.cnt}</td>
47 </tr>
48 </c:forEach>
49 </table>
50 <br>
51 <a href="insertBoard.jsp">새글 등록</a>
```


서버 실행 보기



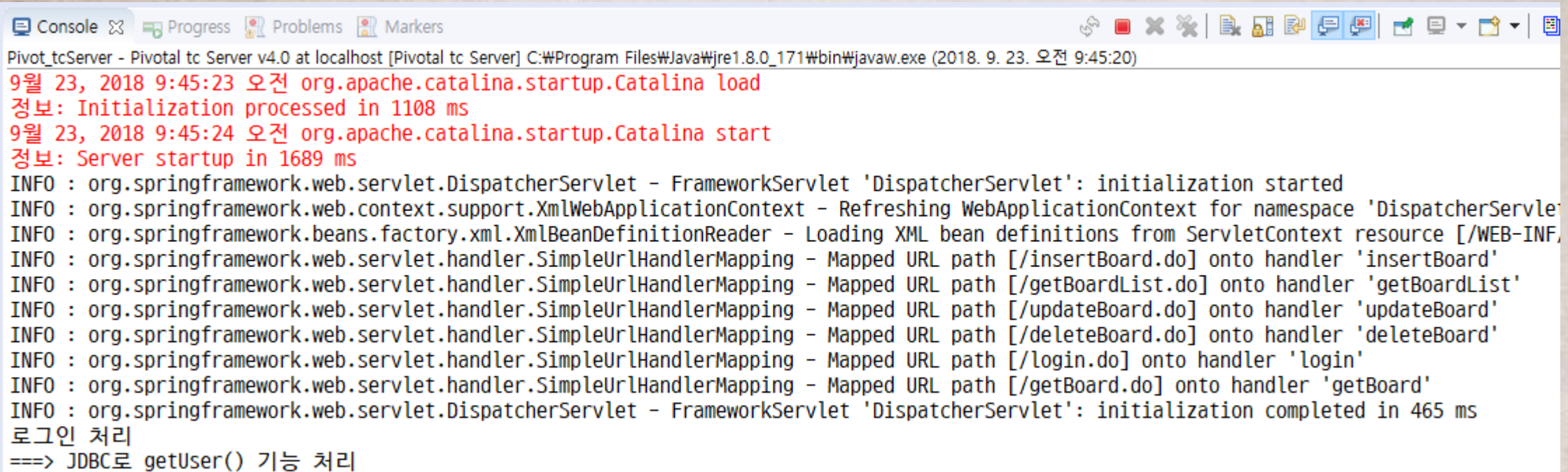
글 목록

환영합니다. [Logout](#)

제목					검색
번호	제목	작성자	등록일	조회수	
5	글 등록	글 등록>	2018-09-10	0	
4	JDBD 테스트2	관리자>	2018-05-27	0	
3	JDBD 테스트	관리자>	2018-05-27	0	
2	임시 제목	홍길동>	2018-05-12	0	
1	가입인사	관리자>	2018-04-08	0	

[새글 등록](#)

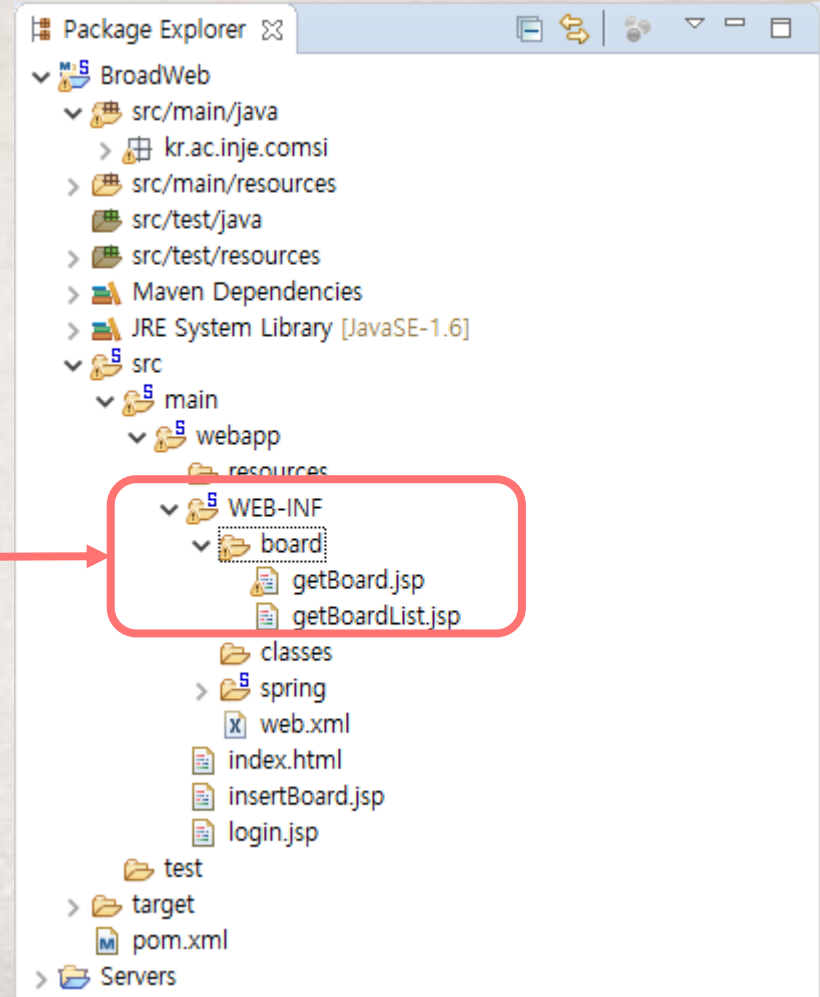
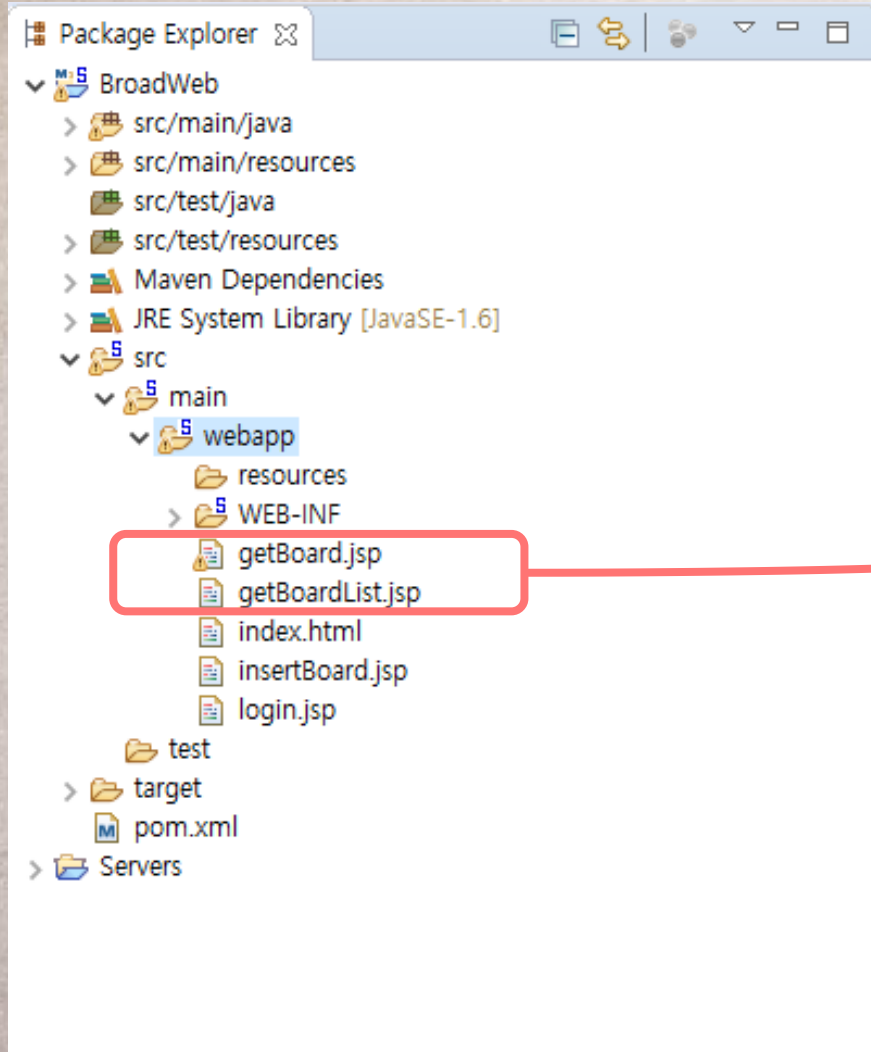
등록된 Controller로가 HandlerMapping로 설정



```
Console Progress Problems Markers
Pivot_tcServer - Pivotal tc Server v4.0 at localhost [Pivotal tc Server] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 9. 23. 오전 9:45:20)
9월 23, 2018 9:45:23 오전 org.apache.catalina.startup.Catalina load
정보: Initialization processed in 1108 ms
9월 23, 2018 9:45:24 오전 org.apache.catalina.startup.Catalina start
정보: Server startup in 1689 ms
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'DispatcherServlet': initialization started
INFO : org.springframework.web.context.support.XmlWebApplicationContext - Refreshing WebApplicationContext for namespace 'DispatcherServlet'
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from ServletContext resource [/WEB-INF/]
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/insertBoard.do] onto handler 'insertBoard'
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/getBoardList.do] onto handler 'getBoardList'
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/updateBoard.do] onto handler 'updateBoard'
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/deleteBoard.do] onto handler 'deleteBoard'
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/login.do] onto handler 'login'
INFO : org.springframework.web.servlet.handler.SimpleUrlHandlerMapping - Mapped URL path [/getBoard.do] onto handler 'getBoard'
INFO : org.springframework.web.servlet.DispatcherServlet - FrameworkServlet 'DispatcherServlet': initialization completed in 465 ms
로그인 처리
==> JDBC로 getUser() 기능 처리
```

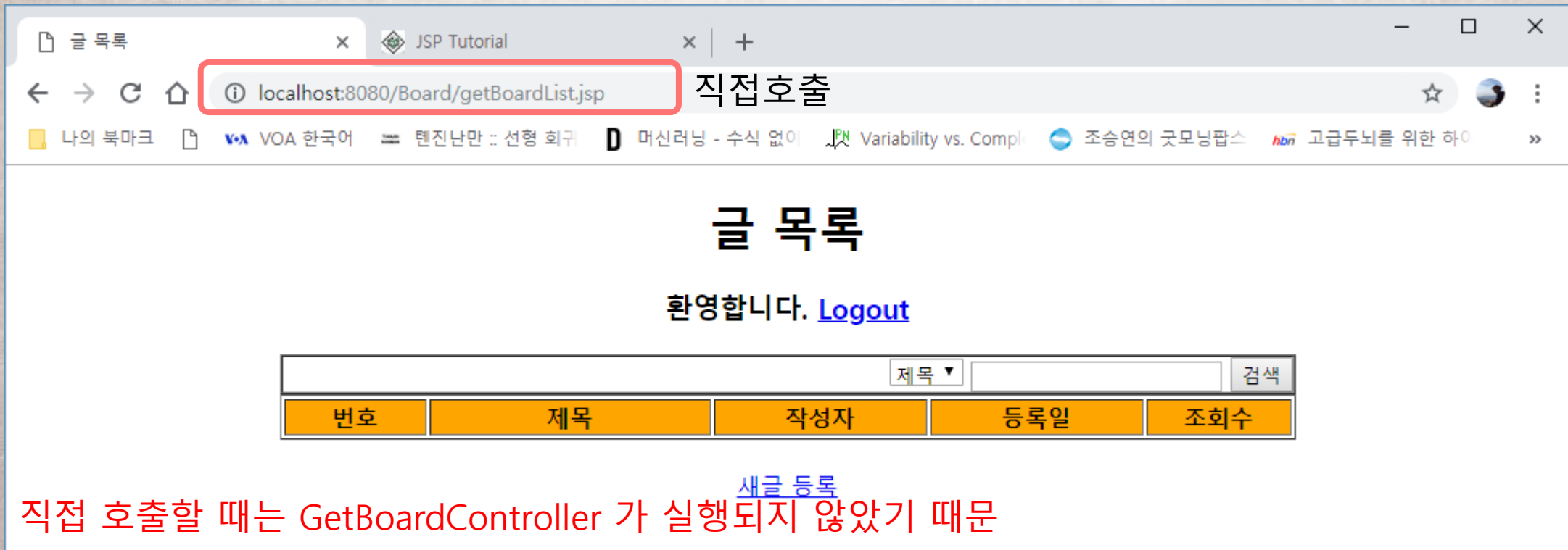

6.9 ViewResolver 활용하기

- View(JSP만) 파일들 이동 – WEB-INF/board 폴더 생성



ViewResolver

- 컨트롤러의 처리결과를 보여줄 View 구현체
- ViewResolver를 이용하면 클라이언트로부터 직접적인 JSP 호출을 차단(오류 발생)
- 대부분의 웹프로젝트에서 ViewResolver 사용은 거의 필수
- ViewResolver 역시 여러가지가 있지만 **JSP를 View로 사용할 경우에는 InternalResourceViewResolver**를 사용한다.



The screenshot shows a web browser window with the address bar containing `localhost:8080/Board/getBoardList.jsp`, which is highlighted with a red box. The page title is "직접호출". The main content area displays "글 목록" (List of Posts) and "환영합니다. [Logout](#)". Below this is a search bar with a dropdown menu set to "제목" (Title) and a "검색" (Search) button. At the bottom, there is a table header with five columns: "번호" (Number), "제목" (Title), "작성자" (Author), "등록일" (Registration Date), and "조회수" (View Count).

직접 호출할 때는 [새글 등록](#) GetBoardController 가 실행되지 않았기 때문

servlet-context.xml에 ViewResolver추가

```
26 <bean id="insertBoard" class="kr.ac.inje.comsi.view.board.InsertBoardController"/></bean>
27 <bean id="updateBoard" class="kr.ac.inje.comsi.view.board.UpdateBoardController"/></bean>
28 <bean id="deleteBoard" class="kr.ac.inje.comsi.view.board.DeleteBoardController"/></bean>
29
30 <!-- ViewResolver 등록 -->
31 <bean id="viewResolver"
32     class="org.springframework.web.servlet.view.InternalResourceViewResolver">
33
34     <property name="prefix" value="/WEB-INF/board/"></property>
35     <property name="suffix" value=".jsp"></property>
36 </bean>
37 |
```

- prefix + **파일이름** + suffix 이름의 JSP 파일로 보낸다.
 - 파일이름은 controller에서 넘겨줌

LoginController 클래스 수정

- ViewResolver를 적용할 때, ModelAndView 객체에 저장되는 View 이름은 ViewResolver를 고려하여 등록해야 한다.

```
31 // 3. 화면 이동
32 ModelAndView mav = new ModelAndView();
33 if (user != null) {
34     mav.setViewName("getBoardList.do");
35 }else {
36     mav.setViewName("login.jsp");
37 }
38 return mav;
```

기존 뷰페이지 설정



```
31 // 3. 화면 이동
32 ModelAndView mav = new ModelAndView();
33 if (user != null) {
34     mav.setViewName("redirect:getBoardList.do");
35 }else {
36     mav.setViewName("redirect:login.jsp");
37 }
38 return mav;
```

ViewResolver를 무시하고
바로 뷰페이지로 이동

- View 이름 앞에 **'redirect:'** 붙여 지정하면 ViewResolver를 무시하고 리다이렉트 됨

GetBoardListController 클래스 수정

```
28 // 3. 검색 결과를 ModelAndView에 저장하고 목록 화면을 리턴
29 ModelAndView mav = new ModelAndView();
30 mav.addObject("boardList", boardList);
31 mav.setViewName("getBoardList"); .jsp를 제거
32 return mav;
```

- 게시물을 검색하고 나면 getBoardList.jsp가 실행
 - 이때, 확장자가 '.jsp'를 생략하면 ViewResolver가 접두사와 접미사를 적절히 할당하여 '/WEB-INF/board/getBoardList.jsp'파일을 실행한다.
- InternalResourceViewResolver를 등록했을 때는 모든 View 이름에 확장자를 제거해야 한다.
- 또한, 확장자가 '.do'인 요청은 앞에 'redirect:'를 붙여서 ViewResolver가 동작하지 않도록 해야 한다.

GetBoardController 클래스 수정

```
30 // 3. 검색 결과를 ModelAndView에 저장하고 목록 화면을 리턴
31 ModelAndView mav = new ModelAndView();
32 mav.addObject("board", board);
33 mav.setViewName("getBoard");
34 return mav; //
```

.jsp를 제거

기타 Controller 클래스

- InsertBoardController, UpdateBoardController, DeleteBoardController 클래스는 모두 'redirect:getBoardList.do'로 수정

```
30 // 3. 삭제하고 목록 화면을 리턴
31 ModelAndView mav = new ModelAndView();
32 mav.setViewName("redirect:getBoardList.do"); 'redirect:' 추가
33 return mav; ///
```

- LogoutController 클래스

```
22 // 2. 세션 종료후, 메인 화면으로 이동
23 ModelAndView mav = new ModelAndView();
24 mav.setViewName("redirect:login.jsp"); 'redirect:' 추가
25 return mav;
```

- Logout처리 후에 로그인 화면으로 이동