박 경 태
comsi.java@gmail.com

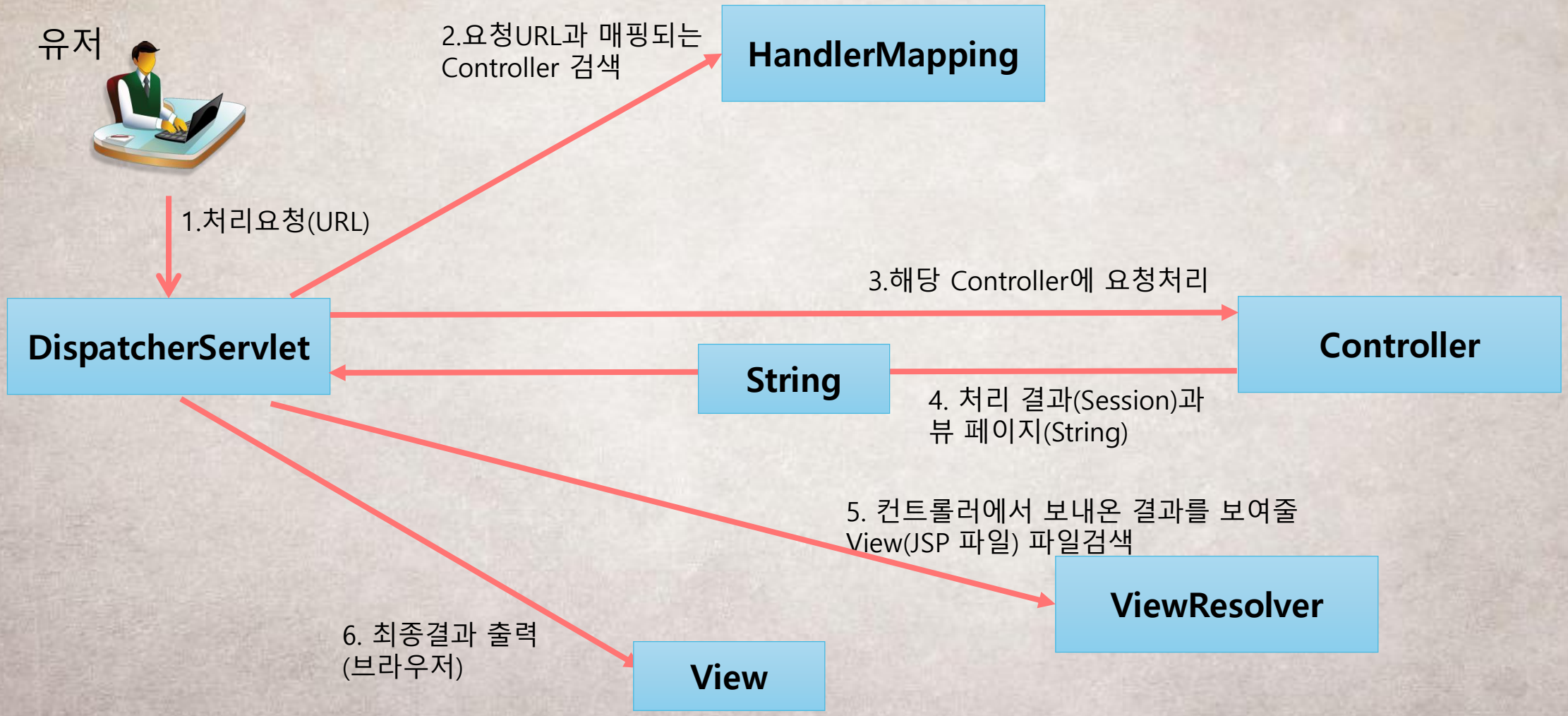
고급 자바 프로그래밍
: STS를 이용한 Spring 프로그래밍

4. MVC를 이용한 웹어플리케이션

4.1 MVC 프레임워크 구조

스프링 MVC의 수행 흐름

스프링 MVC의 클라이언트 요청 처리 과정



스프링 MVC를 이용한 웹어플리케이션

- 스프링 MVC의 주요 구성 요소

구성요소	설명
DispatcherServlet	유일한 서블릿 클래스로서 모든 클라이언트의 요청을 가장 먼저 처리하는 Front Controller
HandlerMapping	클라이언트의 요청(URL)을 어떤 Controller가 처리할지를 결정
Controller	클라이언트의 요청을 처리한 뒤, 필요시 결과를 세션에 저장하고 뷰페이지(JSP) 정보를 넘김
ViewResolver	Controller가 리턴한 View 이름으로 실행할 JSP경로(접두사+view이름+접미사)를 완성한다.

web.xml

servlet-context.xml | BoardDAO.java | *web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                             http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">

    <servlet>
        <display-name>DispatcherServlet</display-name>
        <servlet-name>DispatcherServlet</servlet-name>
        <servlet-class>kr.ac.inje.comsi.view.controller.DispatcherServlet</servlet-class>


    </servlet>
    <servlet-mapping>
        <servlet-name>DispatcherServlet</servlet-name>
        <url-pattern>*.do</url-pattern>
    </servlet-mapping>
</web-app>

```

4.2 MVC 프레임 구현

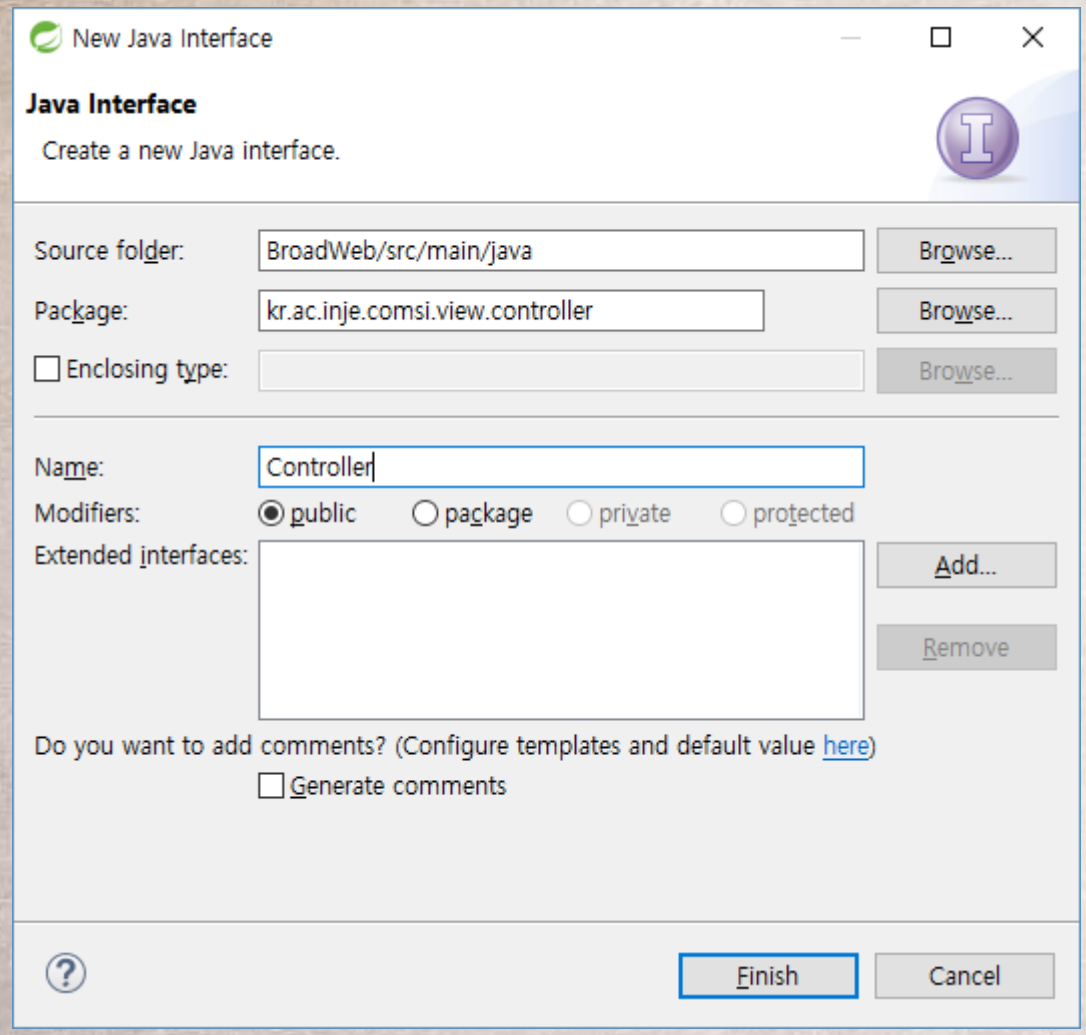
(1) Controller 인터페이스 작성

- DisapatchServlet은 클라이언트의 요청을 가장 먼저 받아들임
- 클라이언트의 요청을 처리하기 위해서 하는 일은 거의 없음

→ 실질적인 요청 처리는 각 Controller가 담당

- 클라이언트의 요청을 받은 DispatcherServlet은 HandlerMapping을 통해 Controller 객체를 검색하고, 검색된 Controller를 실행한다.
- 모든 Controller를 같은 타입으로 관리하기 위한 인터페이스를 만들어야 한다.
  - 즉, 어떤 Controller객체가 검색되더라도 같은 코드로 실행하려면 모든 Controller의 최상위 인터페이스가 필요

src/main/java 폴더에서



Controller interface – 모든 Controller가 사용

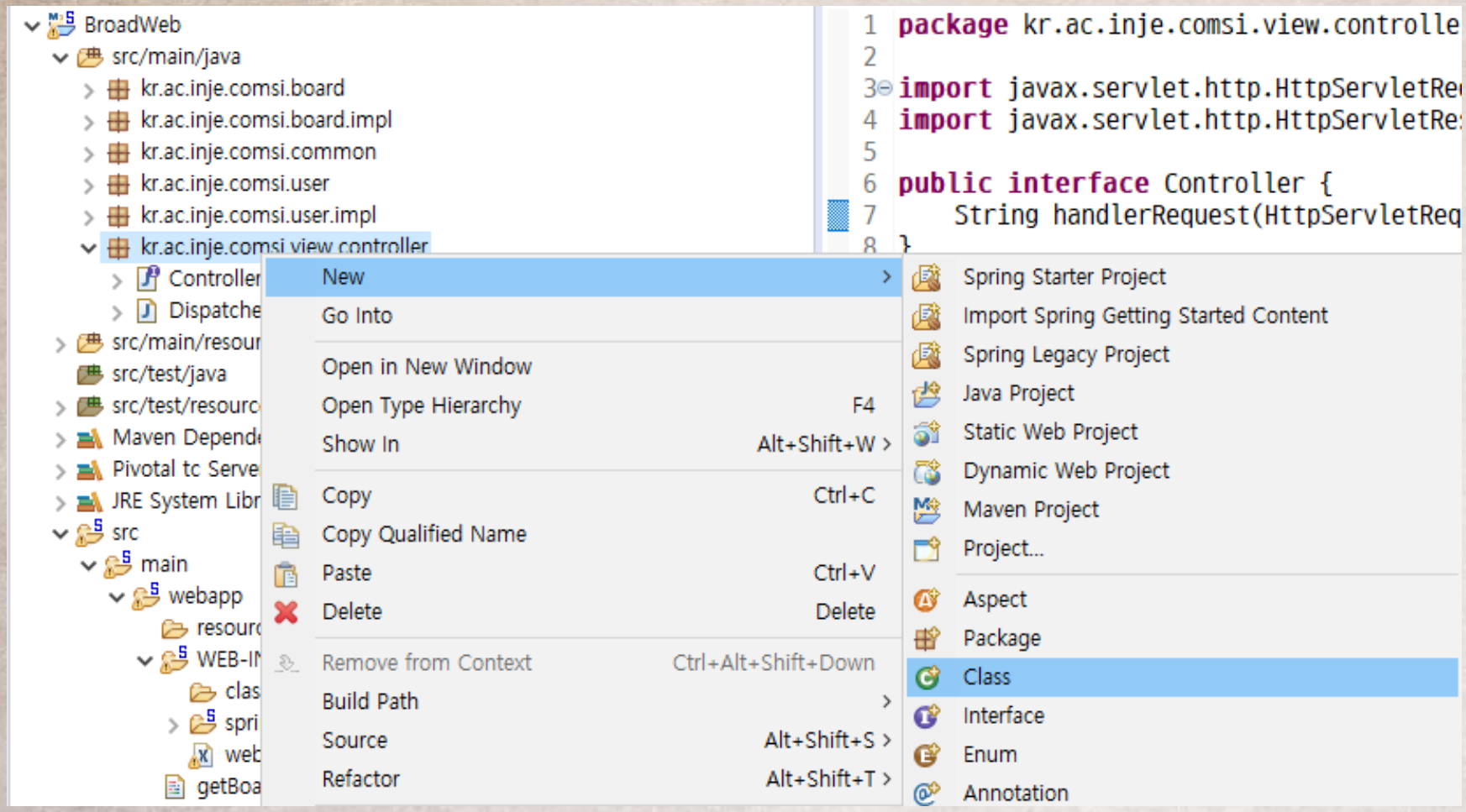
DispatcherServlet.java | Controller.java

```
1  package kr.ac.inje.comsi.view.controller;
2
3  import javax.servlet.http.HttpServletRequest;
4  import javax.servlet.http.HttpServletResponse;
5
6  public interface Controller {
7      String handlerRequest(HttpServletRequest request, HttpServletResponse response);
8  }
9
10
```

'r'을 제거한다(오타).

➔ 이후 모든 Controller 클래스에서도 제외

(2) LoginController 구현



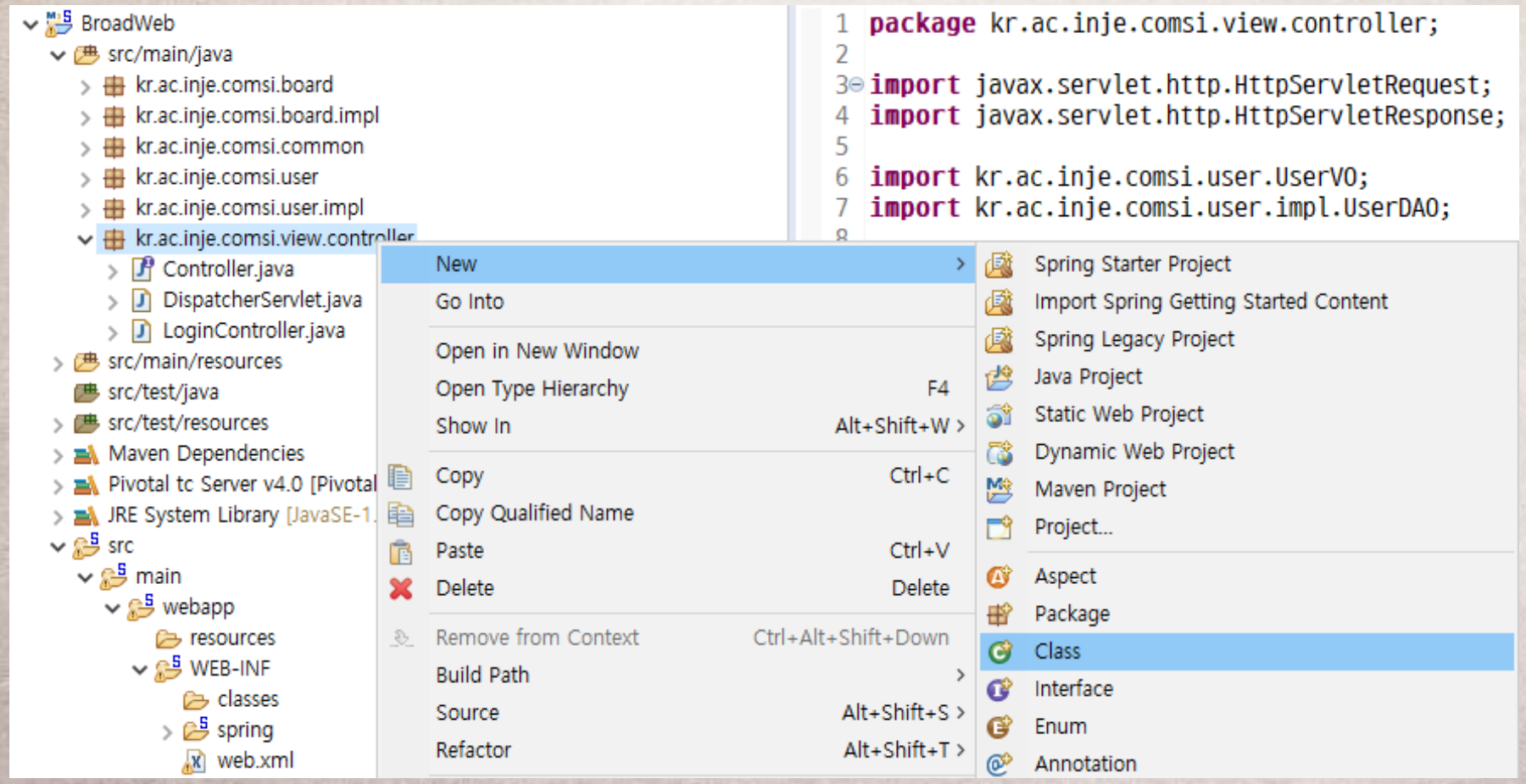
UpdateBoardController.java | GetBoardListController.java | LoginController.java

```java
package kr.ac.inje.comsi.view.user;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import kr.ac.inje.comsi.user.UserVO;
import kr.ac.inje.comsi.user.impl.UserDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class LoginController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request, HttpServletResponse response) {
        System.out.println("로그인 처리");

        // 1. 사용자 입력 정보 추출
        String id = request.getParameter("id");
        String password = request.getParameter("password");

        // 2. DB 연동 처리
        UserVO vo = new UserVO();
        vo.setId(id);
        vo.setPassword(password);

        UserDAO userDAO = new UserDAO();
        UserVO user = userDAO.getUser(vo);

        // 3. 화면 이동
        if (user != null) {
            return "getBoardList.do"; // DispatcherSerlvet에서 받아서 처리
        }else {
            return "login"; // ViewResolver 클래스에서 처리
        }
    }
}
```

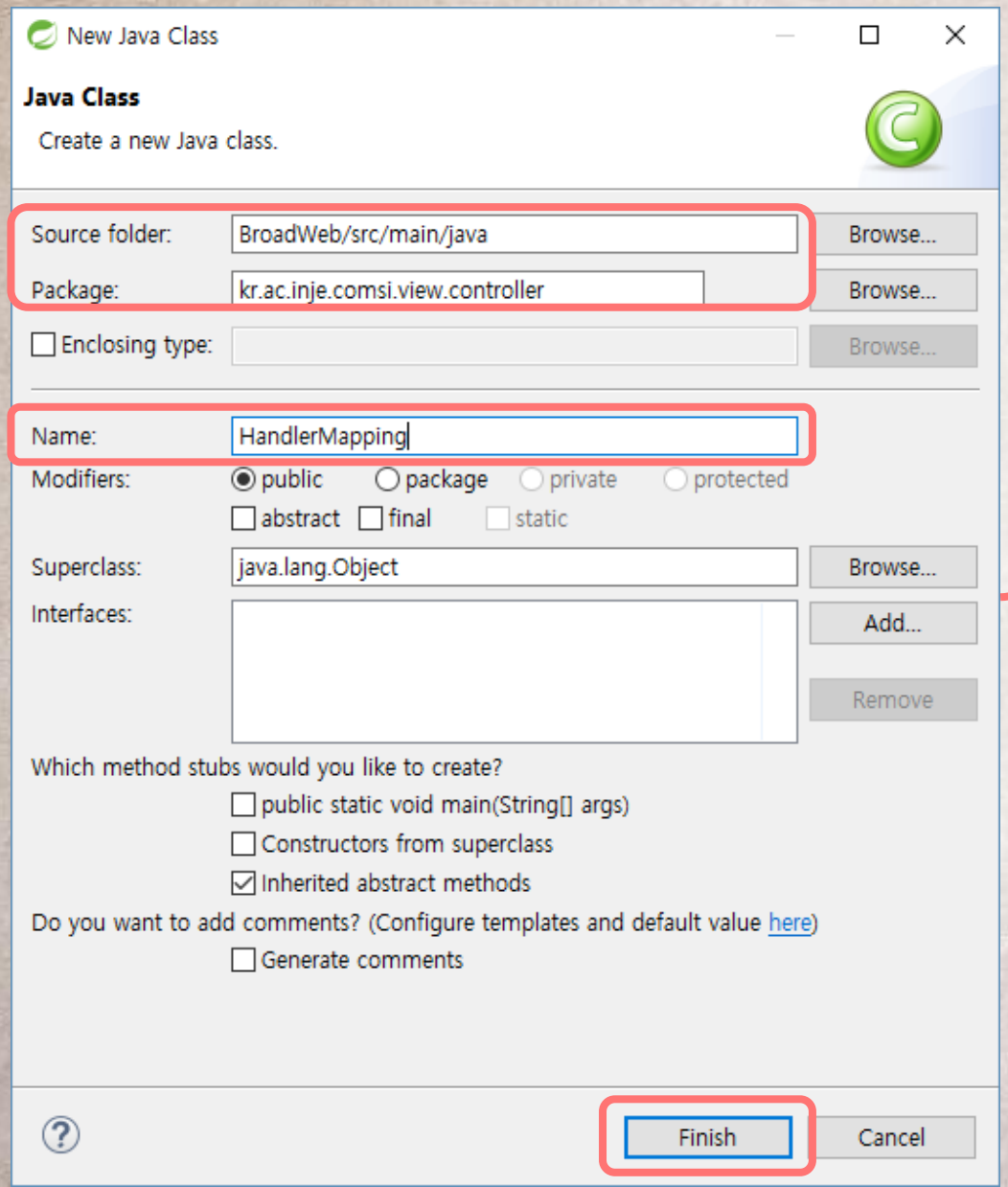
'r'을 제거한다(오타).
➔ 이후 모든 Controller 클래스에서도 제외

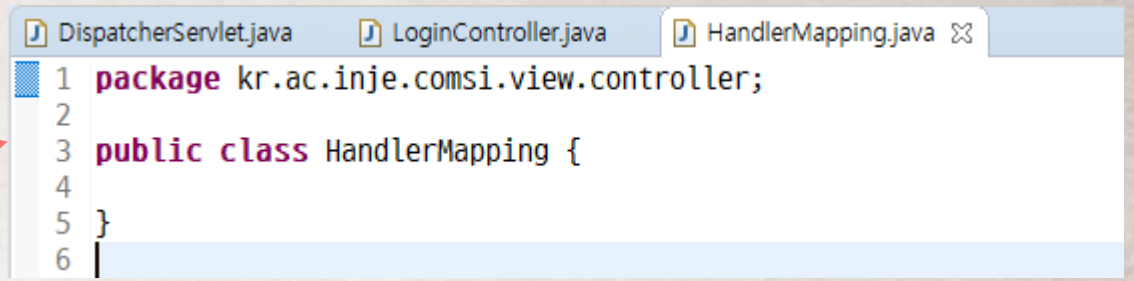
(3) HandlerMapping 클래스 작성

- HandlerMapping은 모든 Controller 객체를 저장(Map)하고 있다가 클라이언트의 요청이 들어오면 처리할 특정 Controller를 검색하는 기능
- DispatcherServlet에서 사용하고 init() 메소드가 호출될 때 단 한 번 생성



이어서…


New Java Class
Java Class
Create a new Java class.
Source folder: BroadWeb/src/main/java
Browse...
Package: kr.ac.inje.comsi.view.controller
Browse...
Enclosing type:
Browse...
Name: HandlerMapping
Modifiers: public package private protected
abstract final static
Superclass: java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel


DispatcherServlet.java
LoginController.java
HandlerMapping.java
1 package kr.ac.inje.comsi.view.controller;
2
3 public class HandlerMapping {
4
5 }
6

HandlerMapping.java 클래스

- HandlerMapping은 **Map 타입 컬렉션** 멤버변수를 가지고 게시판 프로그램에 필요한 모든 Controller객체 등록하고 관리

DispatcherServlet.java | LoginController.java | HandlerMapping.java

```
1  package kr.ac.inje.comsi.view.controller;
2
3  import java.util.HashMap;
4  import java.util.Map;
5
6  public class HandlerMapping {
7      private Map<String, Controller> mappings;
8
9      // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
10     public HandlerMapping() {
11         mappings = new HashMap<String, Controller>();
12         mappings.put("/login.do", new LoginController());
13     }
14
15     // 요청경로에 따른 Controller 얻기
16     public Controller getController(String path) {
17         return mappings.get(path);
18     }
19 }
20
```

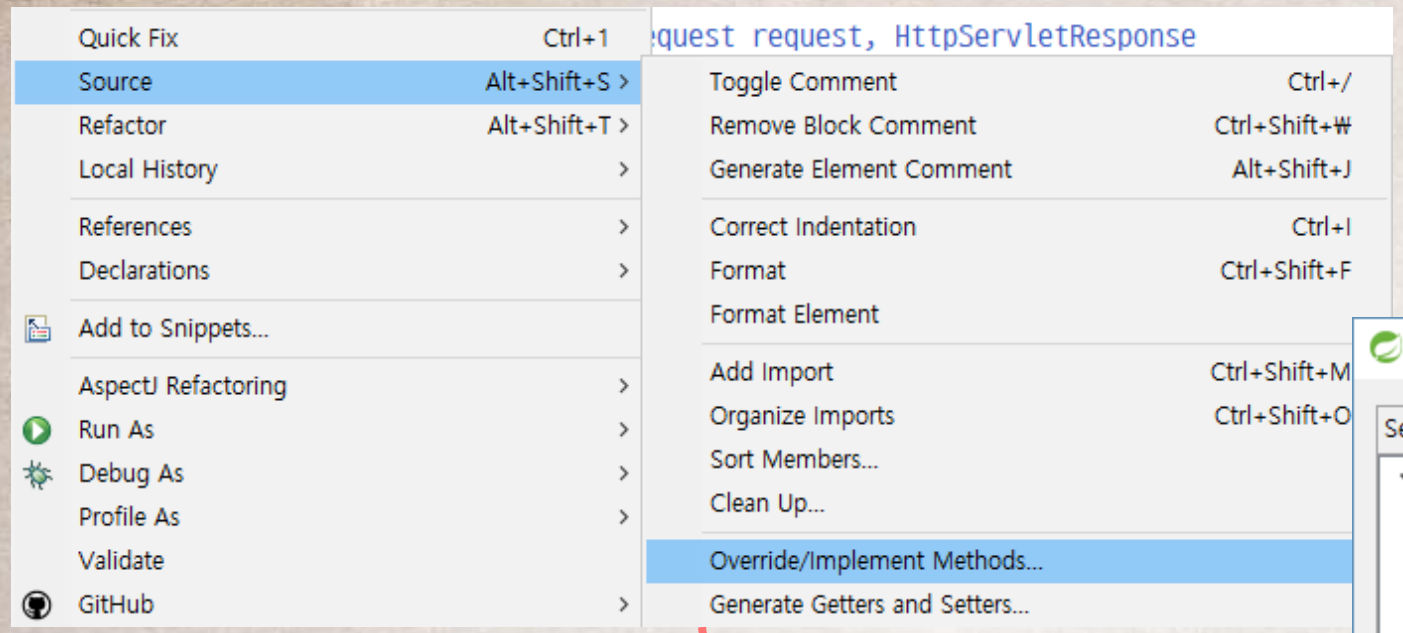
(4) ViewResolver클래스 작성

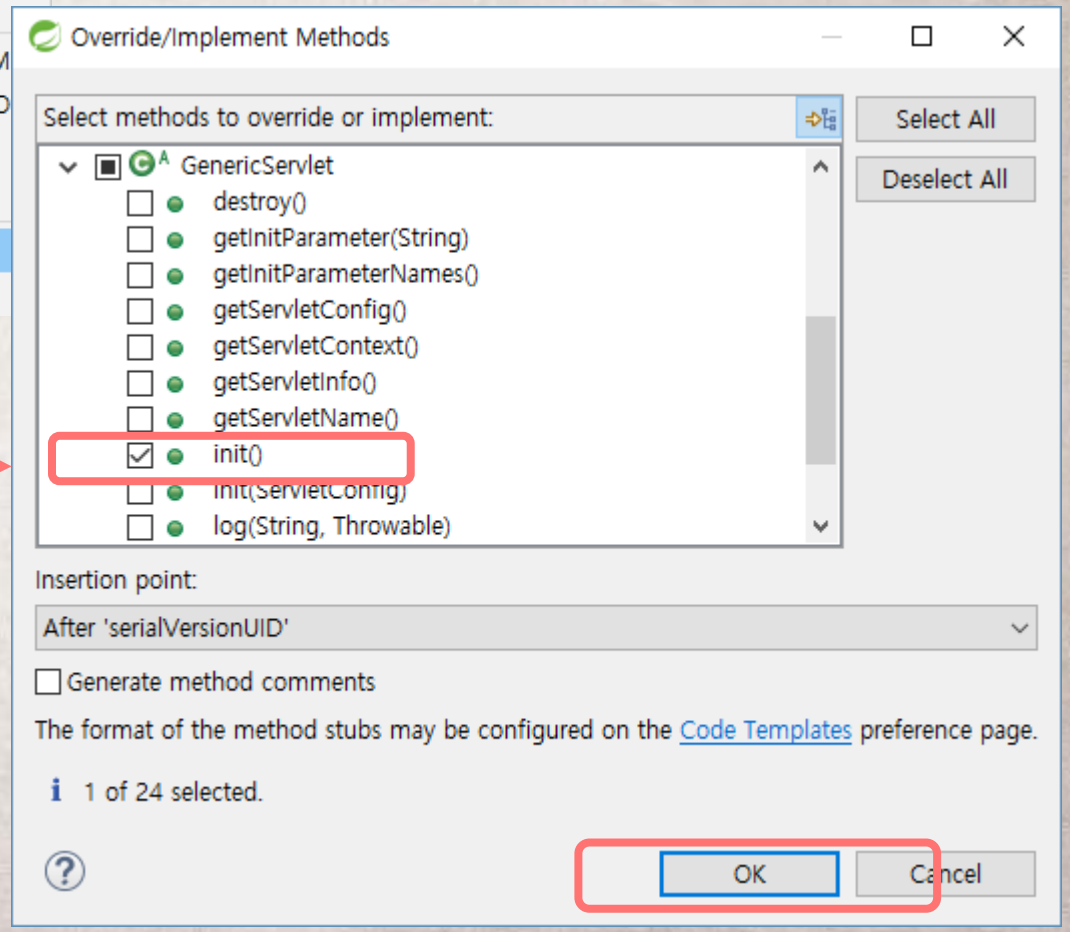
- ViewResolver클래스는 Controller가 리턴한 View에 접두사와 접미사를 결합하여 최종 실행된 View 경로와 파일명을 완성한다.
- DispatcherServlet의 init() 메소드가 호출될 때 단 한 번 생성

DispatcherServlet.java | LoginController.java | HandlerMapping.java | ViewResolver.java

```java
package kr.ac.inje.comsi.view.controller;

public class ViewResolver {
    public String prefix;
    public String suffix;

    public void setPrefix(String prefix) {
        this.prefix = prefix;
    }

    public void setSuffix(String suffix) {
        this.suffix = suffix;
    }

    // prefix - 파일이 있는 폴더
    // viewName - 보내질 파일 이름
    // suffix - 파일 확장자(.jsp)
    public String getView(String viewName) {
        return prefix+viewName + suffix;
    }
}
```

(5) DispatcherServlet 수정





DispatcherServlet 수정

DispatcherServlet.java | LoginController.java | HandlerMapping.java | ViewResolver.java

```java
package kr.ac.inje.comsi.view.controller;

import java.io.IOException;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class DispatcherServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private HandlerMapping handlerMapping;
    private ViewResolver viewResolver;

    @Override
    public void init() throws ServletException {
        handlerMapping = new HandlerMapping();
        viewResolver = new ViewResolver();
        viewResolver.setPrefix("./");
        viewResolver.setSuffix(".jsp");
    }

    public DispatcherServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        process(request, response);
    }
```

이어서…

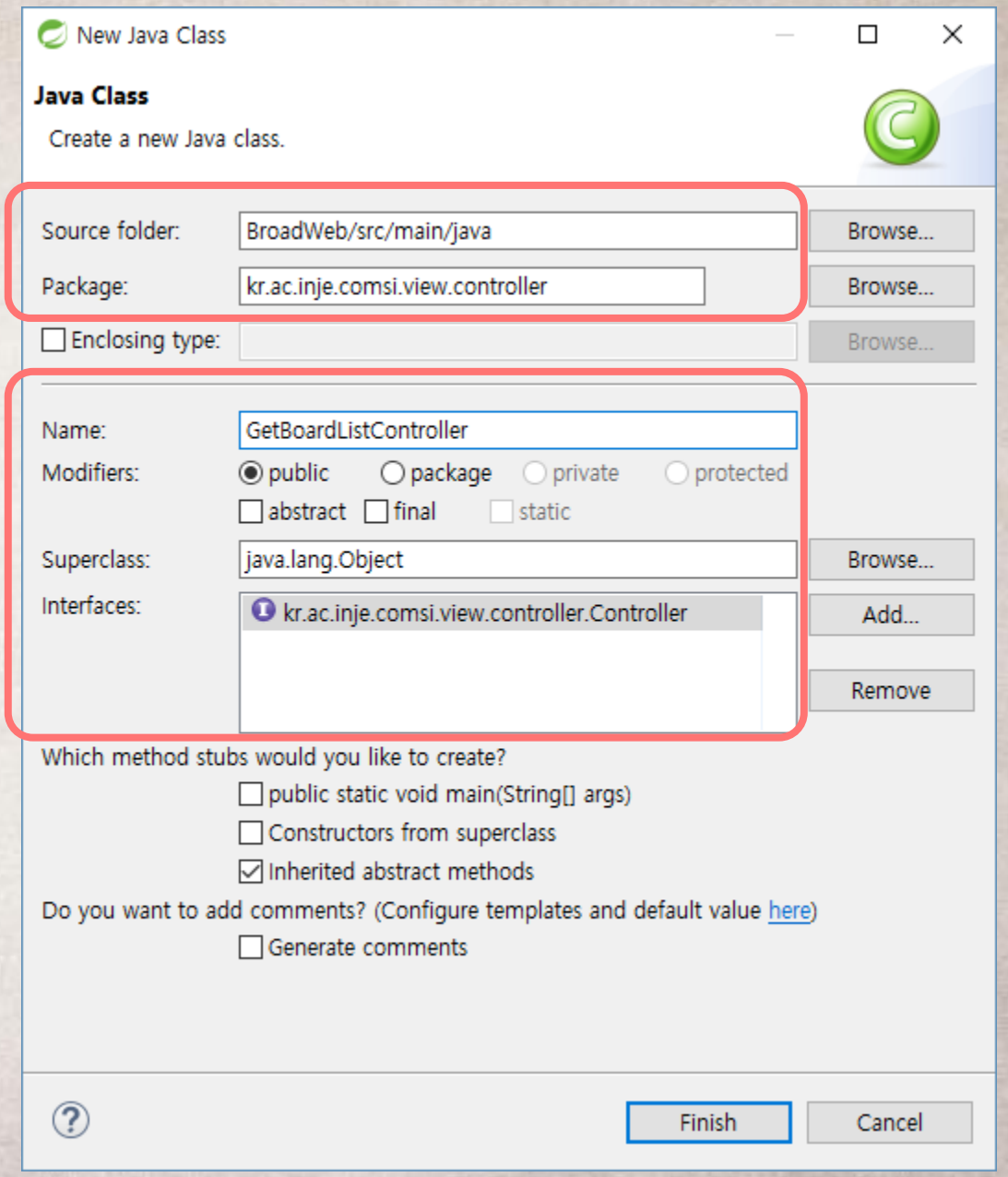
```
34
35      protected void doPost(HttpServletRequest request, HttpServletResponse response)
36              throws ServletException, IOException {
37          request.setCharacterEncoding("UTF-8");
38          process(request, response);
39      }
40
41      private void process(HttpServletRequest request, HttpServletResponse response) throws IOException {
42          // 1. 클라이언트의 요청 path 정보를 추출한다.
43          String uri = request.getRequestURI();
44          String path = uri.substring(uri.lastIndexOf("/"));
45          System.out.println(path);
46
47          // 2. HandlerMapping을 통해 path에 해당하는 Controller를 검색한다.
48          Controller ctrl = handlerMapping.getController(path);
49
50          // 3. 검색된 Controller를 실행한다.
51          String viewName = ctrl.handlerRequest(request, response);
52
53          // 4. ViewResolver를 통해 viewName에 해당하는 화면을 검색한다.
54          String view = null;
55          if (!viewName.contains(".do")) {
56              view = viewResolver.getView(viewName);
57          }else {
58              view = viewName;
59          }
60
61          // 5. 검색된 화면으로 이동한다.
62          response.sendRedirect(view);
63      }
64  }
```

'r'을 제거한다(오타).

4.3 MVC 프레임워크 적용

(1) 글 목록 검색 구현

- Controller 인터페이스를 구현한 GetBoardListController 클래스 작성



GetBoardListController 클래스

LogoutController.java | GetBoardController.java | UpdateBoardController.java | GetBoardListController.java

```java
package kr.ac.inje.comsi.view.board;

import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.board.impl.BoardDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class GetBoardListController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("글 목록 검색 처리");

        // 1. 사용자 입력 정보 추출(검색 기능은 나중에~~)
        // 2. DB 연동 처리
        BoardVO vo = new BoardVO();
        BoardDAO boardDAO = new BoardDAO();
        List<BoardVO> boardList = boardDAO.getBoardList(vo);

        // 3. 검색 결과를 세션에 저장하고 목록 화면을 리턴
        HttpSession session = request.getSession();
        session.setAttribute("boardList", boardList);
        return "getBoardList";
    }
}
```

'r'을 제거한다(오타).
➔ 이후 모든 Controller 클래스에서도 제외

새로운 Controller를 추가했으므로 HandlerMapping 수정

DispatcherServlet.java | HandlerMapping.java | GetBoardListController.java

```java
package kr.ac.inje.comsi.view.controller;

import java.util.HashMap;

public class HandlerMapping {
    private Map<String, Controller> mappings;

    // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
    public HandlerMapping() {
        mappings = new HashMap<String, Controller>();
        mappings.put("/login.do", new LoginController());
        mappings.put("/getBoardList.do", new GetBoardListController());
    }

    // 요청경로에 따른 Controller 얻기
    public Controller getController(String path) {
        return mappings.get(path);
    }
}
```

GetBoardListController 추가

(2) 글 상세보기 구현 – GetBoardController 클래스

LogoutController.java | GetBoardController.java

```java
package kr.ac.inje.comsi.view.board;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.board.impl.BoardDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class GetBoardController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("글 상세 조회 처리");

        // 1. 검색할 게시글 번호 추출
        String seq = request.getParameter("seq");

        // 2. DB 연동 처리
        BoardVO vo = new BoardVO();
        vo.setSeq(Integer.parseInt(seq));

        BoardDAO boardDAO = new BoardDAO();
        BoardVO board = boardDAO.getBoard(vo);

        // 3. 검색 결과를 세션에 저장하고 목록 화면을 리턴
        HttpSession session = request.getSession();
        session.setAttribute("board", board);
        return "getBoard"; //
    }
}
```

새로운 Controller를 추가했으므로 HandlerMapping 수정

HandlerMapping.java | DispatcherServlet.java | GetBoardController.java | GetBoardListController.java

```
1  package kr.ac.inje.comsi.view.controller;
2
3  import java.util.HashMap;
5
6  public class HandlerMapping {
7      private Map<String, Controller> mappings;
8
9      // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
10     public HandlerMapping() {
11         mappings = new HashMap<String, Controller>();
12         mappings.put("/login.do", new LoginController());
13         mappings.put("/getBoardList.do", new GetBoardListController());
14         mappings.put("/getBoard.do", new GetBoardController());
15     }
16
17     // 요청경로에 따른 Controller 얻기
18     public Controller getController(String path) {
19         return mappings.get(path);
20     }
21 }
22
```

GetBoardController 추가

(3) 글 수정 구현 – UpdateBoardController 클래스

LogoutController.java | GetBoardController.java | UpdateBoardController.java

```java
package kr.ac.inje.comsi.view.board;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.board.impl.BoardDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class UpdateBoardController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("글 수정 처리");

        // 1. 사용자 입력한 정보 추출
        String title = request.getParameter("title");
        String content = request.getParameter("content");
        String seq = request.getParameter("seq");

        // 2. DB 연동 처리
        BoardVO vo = new BoardVO();
        vo.setTitle(title);
        vo.setContent(content);
        vo.setSeq(Integer.parseInt(seq));

        BoardDAO boardDAO = new BoardDAO();
        boardDAO.updateBoard(vo);

        // 3. 화면을 리턴
        return "getBoardList.do"; //
    }
}
```

새로운 Controller를 추가했으므로 HandlerMapping 수정

HandlerMapping.java | DispatcherServlet.java | GetBoardController.java | UpdateBoardController.java

```
1  package kr.ac.inje.comsi.view.controller;
2
3  import java.util.HashMap;
5
6  public class HandlerMapping {
7      private Map<String, Controller> mappings;
8
9      // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
10     public HandlerMapping() {
11         mappings = new HashMap<String, Controller>();
12         mappings.put("/login.do", new LoginController());
13         mappings.put("/getBoardList.do", new GetBoardListController());
14         mappings.put("/getBoard.do", new GetBoardController());
15         mappings.put("/updateBoard.do", new UpdateBoardController());
16     }
17
18     // 요청경로에 따른 Controller 얻기
19     public Controller getController(String path) {
20         return mappings.get(path);
21     }
22 }
23
```

UpdateBoardController 추가

(4)글 등록 구현 – InsertBoardController 클래스

LogoutController.java | GetBoardController.java | DeleteBoardController.java | InsertBoardController.java

```java
package kr.ac.inje.comsi.view.board;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.board.impl.BoardDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class InsertBoardController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("글 등록 처리");

        // 1. 사용자 입력한 정보 추출
        String title = request.getParameter("title");
        String content = request.getParameter("content");
        String writer = request.getParameter("writer");

        // 2. DB 연동 처리
        BoardVO vo = new BoardVO();
        vo.setTitle(title);
        vo.setContent(content);
        vo.setWriter(writer);

        BoardDAO boardDAO = new BoardDAO();
        boardDAO.insertBoard(vo);

        // 3. 화면을 리턴
        return "getBoardList.do"; //
    }
}
```

새로운 Controller를 추가했으므로 HandlerMapping 수정

HandlerMapping.java | DispatcherServlet.java | InsertBoardController.java

```
1  package kr.ac.inje.comsi.view.controller;
2
3  import java.util.HashMap;
5
6  public class HandlerMapping {
7      private Map<String, Controller> mappings;
8
9      // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
10     public HandlerMapping() {
11         mappings = new HashMap<String, Controller>();
12         mappings.put("/login.do", new LoginController());
13         mappings.put("/getBoardList.do", new GetBoardListController());
14         mappings.put("/getBoard.do", new GetBoardController());
15         mappings.put("/updateBoard.do", new UpdateBoardController());
16         mappings.put("/insertBoard.do", new InsertBoardController());
17     }
18
19     // 요청경로에 따른 Controller 얻기
20     public Controller getController(String path) {
21         return mappings.get(path);
22     }
23 }
24
```

InsertBoardController 추가

(5) 글 삭제-DeleteBoardController 클래스

LogoutController.java | GetBoardController.java | DeleteBoardController.java

```java
package kr.ac.inje.comsi.view.board;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import kr.ac.inje.comsi.board.BoardVO;
import kr.ac.inje.comsi.board.impl.BoardDAO;
import kr.ac.inje.comsi.view.controller.Controller;

public class DeleteBoardController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("글 삭제 처리");

        // 1. 삭제할 게시글 번호 추출
        String seq = request.getParameter("seq");

        // 2. DB 연동 처리
        BoardVO vo = new BoardVO();
        vo.setSeq(Integer.parseInt(seq));

        BoardDAO boardDAO = new BoardDAO();
        boardDAO.deleteBoard(vo);

        // 3. 삭제하고 목록 화면을 리턴
        return "getBoardList.do"; //
    }
}
```

새로운 Controller를 추가했으므로 HandlerMapping 수정

HandlerMapping.java | DispatcherServlet.java | InsertBoardController.java | DeleteBoardController.java

```java
package kr.ac.inje.comsi.view.controller;

import java.util.HashMap;

public class HandlerMapping {
    private Map<String, Controller> mappings;

    // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
    public HandlerMapping() {
        mappings = new HashMap<String, Controller>();
        mappings.put("/login.do", new LoginController());
        mappings.put("/getBoardList.do", new GetBoardListController());
        mappings.put("/getBoard.do", new GetBoardController());
        mappings.put("/updateBoard.do", new UpdateBoardController());
        mappings.put("/insertBoard.do", new InsertBoardController());
        mappings.put("/deleteBoard.do", new DeleteBoardController());
    }

    // 요청경로에 따른 Controller 얻기
    public Controller getController(String path) {
        return mappings.get(path);
    }
}
```

DeleteBoardController 추가

(6) 로그아웃 구현-LogoutController 클래스

HandlerMapping.java | DispatcherServlet.java | LogoutController.java

```java
package kr.ac.inje.comsi.view.user;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import kr.ac.inje.comsi.view.controller.Controller;

public class LogoutController implements Controller {

    @Override
    public String handlerRequest(HttpServletRequest request,
                                 HttpServletResponse response) {
        System.out.println("로그아웃 처리");

        // 1. 브라우저와 연결된 세션 객체를 강제 종료
        HttpSession session = request.getSession();
        session.invalidate();
        // 2. 세션 종류후, 메인 화면으로 이동
        return "login";
    }

}
```

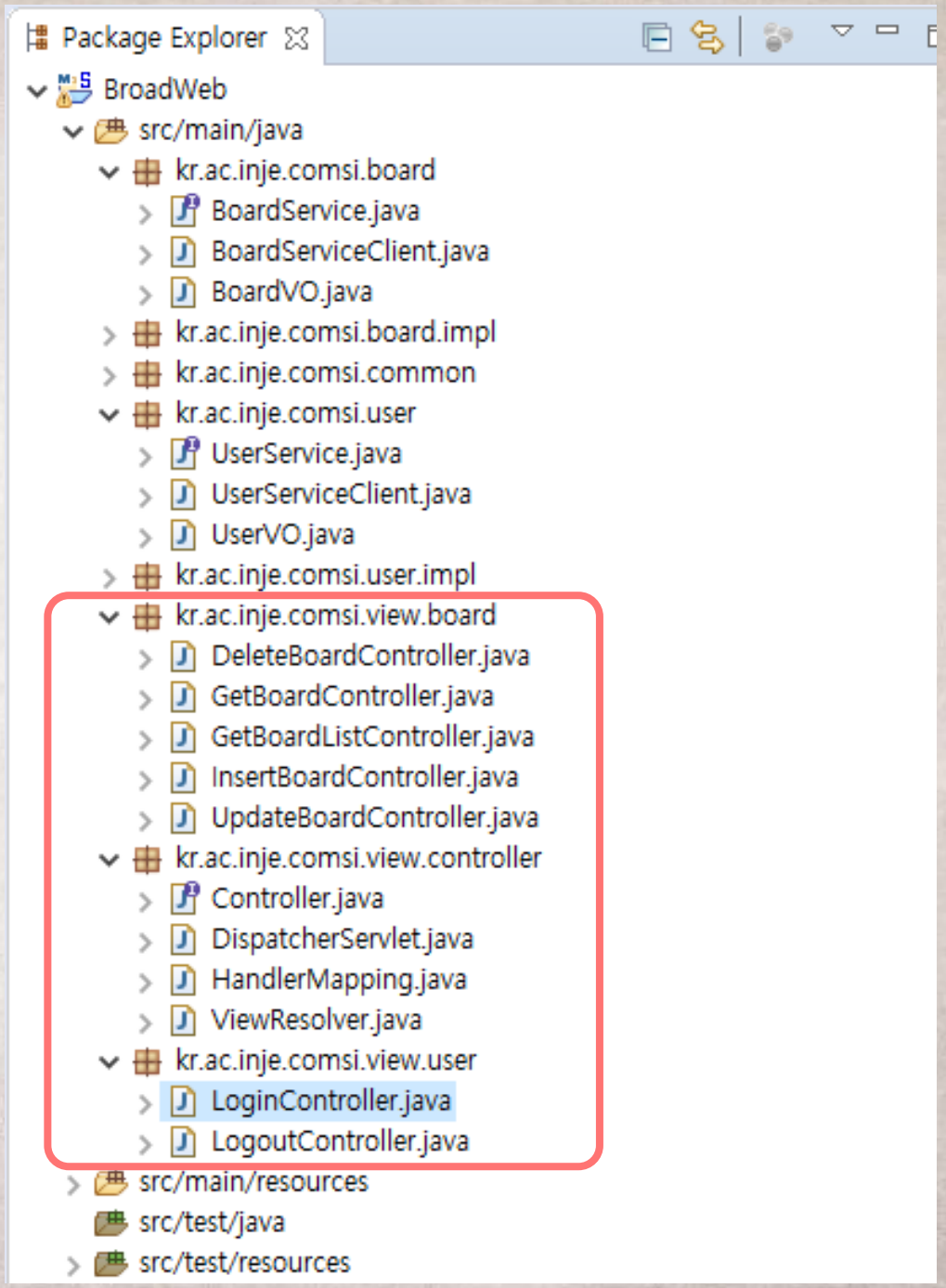
새로운 Controller를 추가했으므로 HandlerMapping 수정

HandlerMapping.java | DispatcherServlet.java | LogoutController.java

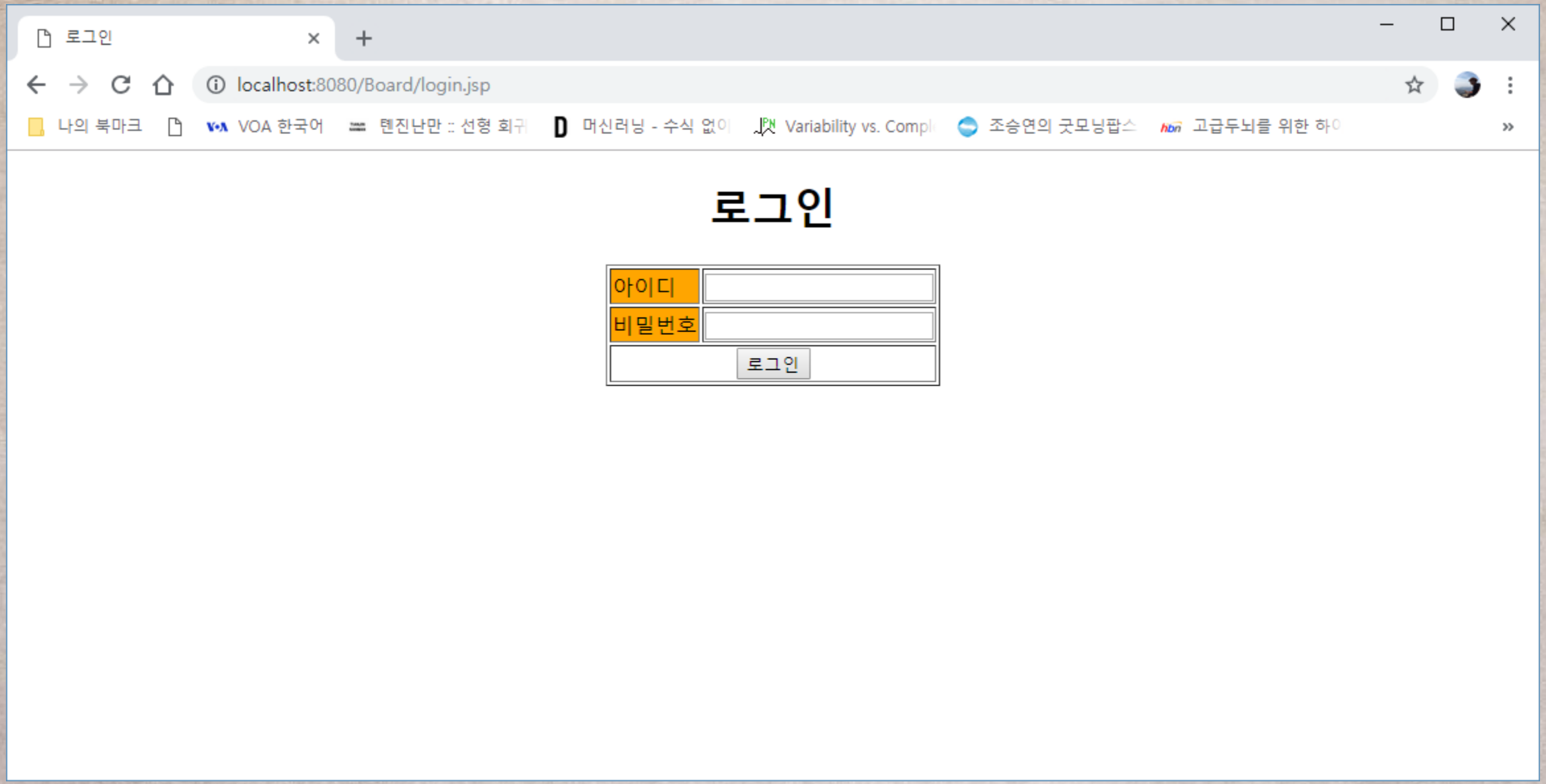
```java
package kr.ac.inje.comsi.view.controller;

import java.util.HashMap;
import java.util.Map;

public class HandlerMapping {
    private Map<String, Controller> mappings;

    // 요청 경로와 해당 컨트롤러를 HashMap으로 생성
    public HandlerMapping() {
        mappings = new HashMap<String, Controller>();
        mappings.put("/login.do", new LoginController());
        mappings.put("/getBoardList.do", new GetBoardListController());
        mappings.put("/getBoard.do", new GetBoardController());
        mappings.put("/updateBoard.do", new UpdateBoardController());
        mappings.put("/insertBoard.do", new InsertBoardController());
        mappings.put("/deleteBoard.do", new DeleteBoardController());
        mappings.put("/logout.do", new LogoutController());
    }

    // 요청경로에 따른 Controller 얻기
    public Controller getController(String path) {
        return mappings.get(path);
    }
}
```

LogoutController 추가

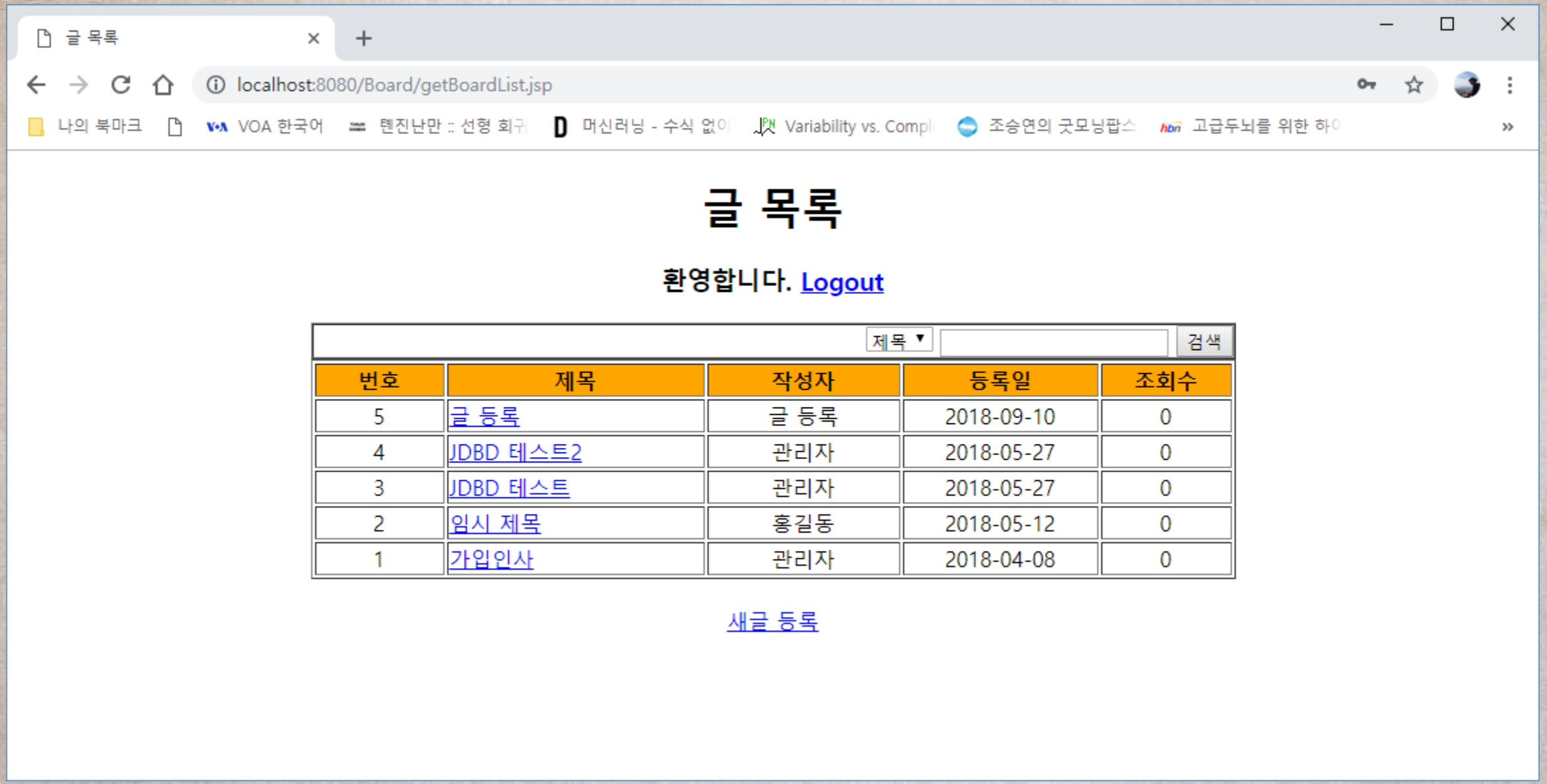
프로젝트 클래스 구조


Package Explorer
BroadWeb
src/main/java
kr.ac.inje.comsi.board
BoardService.java
BoardServiceClient.java
BoardVO.java
kr.ac.inje.comsi.board.impl
kr.ac.inje.comsi.common
kr.ac.inje.comsi.user
UserService.java
UserServiceClient.java
UserVO.java
kr.ac.inje.comsi.user.impl
kr.ac.inje.comsi.view.board
DeleteBoardController.java
GetBoardController.java
GetBoardListController.java
InsertBoardController.java
UpdateBoardController.java
kr.ac.inje.comsi.view.controller
Controller.java
DispatcherServlet.java
HandlerMapping.java
ViewResolver.java
kr.ac.inje.comsi.view.user
LoginController.java
LogoutController.java
src/main/resources
src/test/java
src/test/resources

실행결과


로그인
localhost:8080/Board/login.jsp
나의 북마크
VOA 한국어
로그인
아이디
비밀번호
로그인

글 목록 보기


글 목록
localhost:8080/Board/getBoardList.jsp
나의 북마크
VOA 한국어
머신러닝 - 수식 없이
Variability vs. Compl
조승연의 굿모닝팝스
고급두뇌를 위한 하이
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