

CommandHandler 방식 복습

1) Controller의 doGet 메서드

```
BoardController.java
47 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
48     // TODO Auto-generated method stub
49     String cmd = request.getParameter("cmd");
50
51     if (cmd == null || cmd.equals("")) {
52         cmd = "list";
53     }
54
55     String className = cmds.getProperty(cmd);
56     System.out.println(className);
57     try {
58         //Command command = (Command) Class.forName("cmd.board.ListCommand").newInstance();
59         //ListCommand command = new ListCommand();//cmd=list
60         Command command = (Command) Class.forName(className).newInstance();
61         command.execute(request, response);
62     } catch (InstantiationException e) {
63         // TODO Auto-generated catch block
64         e.printStackTrace();
65     } catch (IllegalAccessException e) {
66         // TODO Auto-generated catch block
67         e.printStackTrace();
68     } catch (ClassNotFoundException e) {
69         // TODO Auto-generated catch block
70         e.printStackTrace();
71     }
}
```

- forname()은 Class object를 반환
- Class object: 해당 클래스의 모든 정보를 담고 있음. 정보를 받은 대상은 클래스 생성 가능. ex. newInstance()

2) el과 jstl

:: jsp파일에서 java 코드를 없애기 위함

- 태그lib 설정

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

- for (object b:arraylist)

```
<c:forEach var="b" items="${list}">
    <tr>
        <td>${b.num}</td>
        <td>${b.writer}</td>
        <td>${b.w_date}</td>
        <td>${b.content}</td>
    </tr>
</c:forEach>
```

1. 게시판 예제

- list.jsp
- WriteFormCommand.java
- writeForm.jsp
- WriteCommand.java *없는 아이디로 글작성하면 오류
- DetailCommand.java (list.jsp에서 각 제목에 링크(detail) 걸고 제목 클릭하면 상세 페이지로 이동하도록.)
- detail.jsp
 - set=>변수 선언 tag
 - 변수 이름은 read. value로 readonly를 넣어준다.
 - 아래에 제목과 내용에 넣어줌
 - 내가 쓴 글은 수정/삭제 가능하게

```

<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script type="text/javascript">
function a(){
    location.href = "${pageContext.request.contextPath }/BoardController?cmd=del&num=${b.num}";
}
</script>
</head>
<body>
<c:if test="${sessionScope.id!=b.writer }">
<c:set var="read">readonly</c:set>
</c:if>
<h3>글 상세페이지</h3>
<a href="${pageContext.request.contextPath }/BoardController?cmd=list">글 목록</a><br>
<form action="${pageContext.request.contextPath }/BoardController?cmd=edit&num=${b.num}" method="post">
<table border="1">
<tr><th>글번호</th><td>${b.num }</td></tr>
<tr><th>제목</th><td><input type="text" value="${b.title }" name="title" ${read }></td></tr>
<tr><th>작성자</th><td>${b.writer }</td></tr>
<tr><th>작성일</th><td>${b.w_date }</td></tr>
<tr><th>내용</th><td><textarea name="content" cols="20" rows="10" ${read }>${b.content }</textarea></td></tr>
<c:if test="${sessionScope.id==b.writer }">
<tr><td colspan="2"><input type="submit" value="수정 "><input type="button" value="삭제 " onclick="a()"></td></tr>
</c:if>
</table>
</form>
</body>
</html>

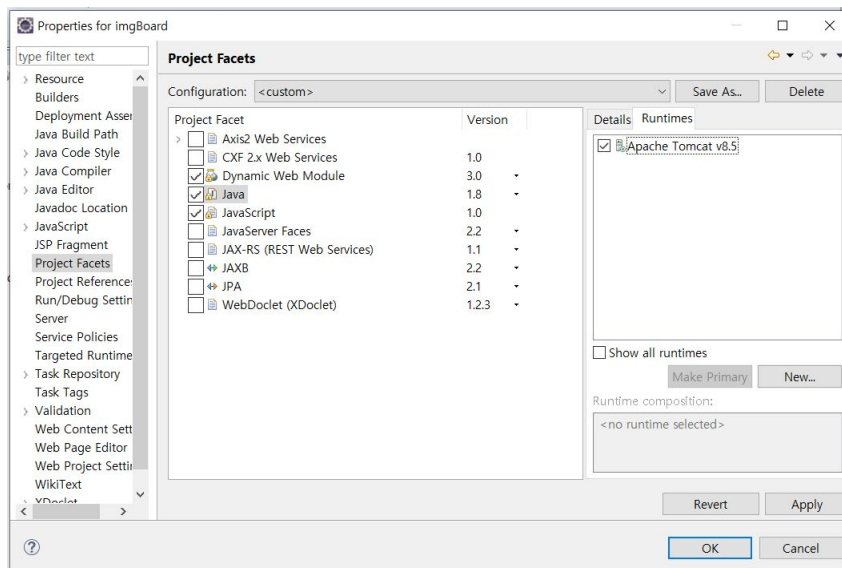
```

- g. EditCommand.java
- h. DelCommand.java

2. 자료실 예제 :: Properties(key, value) 형식 사용 x, 직접 Command 구현

1) 기본 세팅

- a. C:\Users\Playdata\Desktop\workspace\metadata\plugins\org.eclipse.wst.server.core\tmp0\webapps로 가기
*이클립스 내부에 만들면 이미지 파일을 주기적으로 refresh 해줘야 하기 때문에. webapps 안에 생성하기.
 - img 폴더 안 이미지 삭제
 - data 폴더 생성(업로드된 사진 저장용)
- b. 코드 에러 해결 :: 프로젝트 우클릭 => properties



runtime의 tomcat 체크

2) 코드 흐름

- [M,C] fileUploadDownload 패키지 생성
 - FileService.java :: 서비스기능(업로드, 다운로드, 리스트 출력) 정의
 - FileController.java :: 페이지 이동, 서비스 구현
 - cmd==upload ⇒ service의 업로드 기능 호출
 - cmd==down⇒ service의 다운로드 기능 호출
 - cmd==아무거나 ⇒ service의 리스트 출력 기능
- [V] /views/file/ 디렉토리 생성
 - form.jsp :: 파일 업로드 폼

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>파일 업로드</h3>
<form action="{pageContext.request.contextPath }/FileController?cmd=upload" method="post"
    enctype="multipart/form-data">
    title:<input type="text" name="title"><br>
    upload file:<input type="file" name="file"><br>
    <input type="submit" value="upload">
</form>
</body>
</html>
```

- list.jsp :: 파일 목록 (링크로 다운로드)

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>자료실</h3>
<a href="{pageContext.request.contextPath }/views/file/form.jsp">업로드하기</a><br>
<ul>
<c:forEach var="f" items="{flist }">
    <li><a href="{pageContext.request.contextPath }/FileController?cmd=down&fname={f }">{f }</a></li>
</c:forEach>
</ul>
</body>
</html>
```

- CF. ul과 li 태그

UL태그, OL태그, DL태그, LI태그 사용법

태그	비고
UL	글머리 기호를 앞에 붙여 목록을 만드는 형식
OL	번호를 앞에 붙여 목록을 만드는 형식
DL	기호 없이 문단으로 목록을 만드는 형식
LI	UI와 OL내부에서 활용되는 태그

1. UL태그는 앞에 글머리 기호를 붙여 목록을 만드는 방식으로 순서가 필요없는 목록을 만들때 주로 사용합니다.
2. OL태그는 주로 순서가 필요한 목록을 만들때 주로 사용하는데 `ol type="속성"`을 줘 다양한 형태의 목록을 만들 수 있습니다.
3. DL는 앞에 기호없이 문단의 여백으로 목록을 만드는 방식입니다. 시작은 `<DL>` 제목은 `<DT>` 내용은 `<DD>`태그로 설정합니다.
4. LI태그는 UI와 OL태그의 안에 들어가는 태그입니다. 이 **LI태그가 UL와 OL태그의 실질적인 내용 역할**을 합니다.

3. Facebook 예제

1) 오라클 db 생성

- 생성 이유
 - 타이틀 값을 함께 자료와 저장하기 위해 db가 필요
 - db에는 파일명과 경로만 저장

- 테이블 생성

```
create table img_board(  
  num number primary key,  
  writer varchar2(20) not null,  
  title varchar2(50) not null,  
  img varchar2(100) not null  
);
```

- 시퀀스 생성

```
create sequence seq_img_board;
```

2) [M,C] /src/img_board 패키지 생성

- ImgBoard (DTO) 클래스
- dao 인터페이스
- daoImpl
- service 인터페이스
- serviceImpl
- FileUpload.java :: 파일업로드 서비스 기능 정의

```

package img_board;

import java.io.File;

import javax.servlet.http.HttpServletRequest;

import com.oreilly.servlet.MultipartRequest;
import com.oreilly.servlet.multipart.DefaultFileRenamePolicy;

public class FileUpload {
    private String path = "C:\\Users\\Playdata\\Desktop\\workspace\\.metadata\\.plugins\\org.eclipse.wst.server.core\\tmp0\\webapps\\img\\";
    private String webPath = "/img/";

    public ImgBoard upload(HttpServletRequest request) {
        ImgBoard ib = null;
        int maxSize = 10 * 1024 * 1024; // 전송파일의 최대 크기
        String format = "UTF-8";
        try {
            //MultipartRequest를 이용한 파일업로드 방법
            // ↓ request 객체,           ↓ 저장될 서버 경로,           ↓ 파일 최대 크기,           ↓ 인코딩 방식,           ↓ 같은 이름의 파일명 방지 처리
            // (HttpServletRequest request, String saveDirectory, int maxPostSize, String encoding, FileRenamePolicy policy)
            // 아래와 같이 MultipartRequest를 생성만 해주면 파일이 업로드 된다.(파일 자체의 업로드 완료)
            MultipartRequest multi = new MultipartRequest(request, path, maxSize, format,
                new DefaultFileRenamePolicy());

            // MultipartRequest로 전송받은 데이터를 불러온다.
            // enctype을 "multipart/form-data"로 선언하고 submit한 데이터들은 request객체가 아닌 MultipartRequest객체로 불러와야 한다.
            String writer = multi.getParameter("writer");
            String title = multi.getParameter("title");// 입력양식
            File f = multi.getFile("file");// 파일업로드
            String img = webPath + f.getName();
            ib = new ImgBoard(0, writer, title, img);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ib;
    }
}

```

- ImgBoardController :: 페이지 이동
 - cmd==write
 - :: FileUpload 객체 생성
 - :: 파일 업로드 기능 호출
 - cmd==모든 경우
 - :: list=service.getAll()
 - :: list.jsp로 forward 이동

3) [V] /views/imgboard 디렉토리 생성

- list.jsp *enctype="multipart/form-data" 파일업로드용


```

<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>얼굴 책</h3>
<form action="${pageContext.request.contextPath }/ImgBoardController?cmd=write" method="post"
    enctype="multipart/form-data">
<table border="1">
<tr><th>title</th><td><input type="text" name="title"></td></tr>
<tr><th>writer</th><td><input type="text" name="writer"></td></tr>
<tr><th>img</th><td><input type="file" name="file"></td></tr>
<tr><td colspan="2"><input type="submit" value="upload"></td></tr>
</table>
</form>
<c:forEach var="ib" items="${list }">
<table border="1">
<tr><th>num</th><td>${ib.num }</td></tr>
<tr><th>title</th><td><input type="text" name="title" value="${ib.title }" readonly></td></tr>
<tr><th>writer</th><td><input type="text" name="writer" value="${ib.writer }" readonly></td></tr>
<tr><th>img</th><td></td></tr>
<tr><th>댓글</th><td>내용:<input type="text" name="rep"><br/>
<input type="button" value="댓글 작성"></td></tr>
</table>
</c:forEach>
</body>
</html>

```

- 연습문제
 - : 이미지 마다 댓글달고 해당 이미지 밑에 출력되게끔
- 테이블 생성 (img_board_rep)
 - 댓글번호
 - 원문번호
 - 작성자
 - 내용