

JSP

7장. 파일 업로드와 간이 웹 하드 구현

이번 장에서 공부할 것

- 사용자로부터 파일을 업로드 받는 방법
- 이를 이용하여 기본적인 기능만을 제공하는 간이 웹 하드 프로그램을 작성
 - 여러 개의 모듈로 구성된 데이터베이스 응용 프로그램을 작성하는 과정을 파악
- 학습내용
 - 파일 업로드
 - 웹 하드 프로그램을 위한 테이블 생성과 메인 페이지
 - 파일 추가와 삭제 기능

7.1 파일 업로드

- 준비 작업
 - 파일 업로드 기능을 구현한 라이브러리 파일 설치
 - 우리는 사람들이 가장 많이 이용하는 O'Reilly의 COS 라이브러리를 이용
 - <http://www.servlets.com/cos/> 에서 cos-20.08.zip 다운로드
 - 여기에서 우리가 필요한 것은 cos.jar 파일 뿐이므로 압축을 해제한 뒤, 이 파일을 프로젝트 폴더 > WebContent > WEB-INF > lib 폴더에 드래그

7.1 파일 업로드

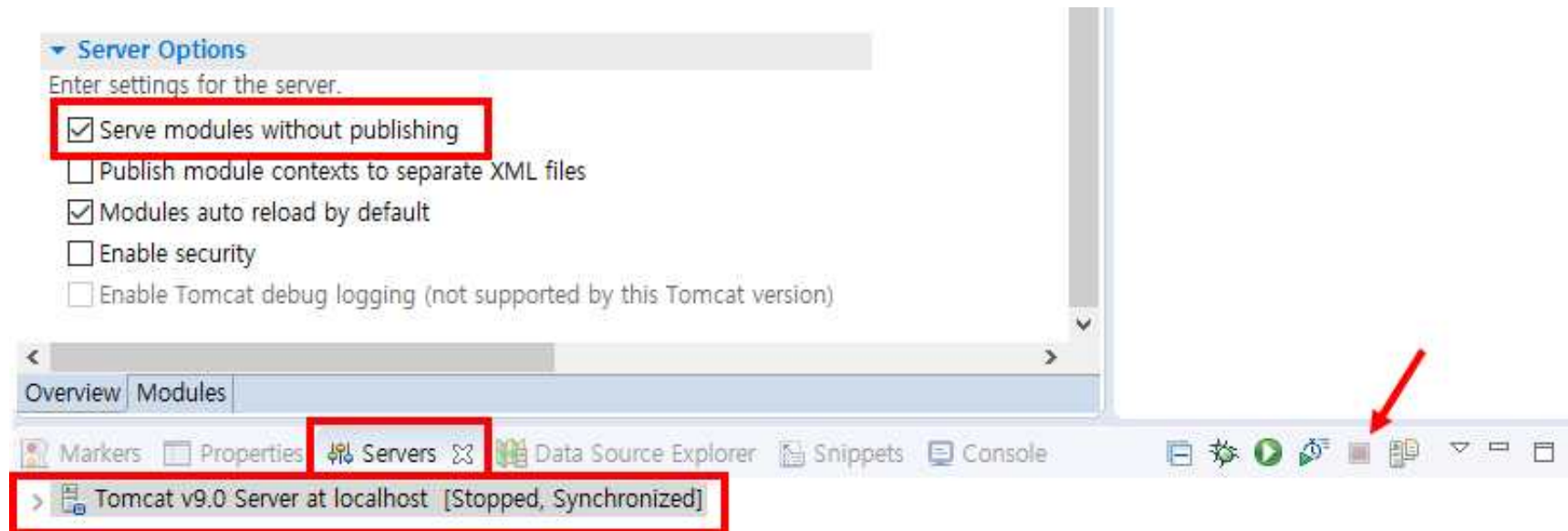
■ 준비 작업

■ 이클립스의 퍼블리싱 기능을 해제

- 퍼블리싱 기능 : 웹 애플리케이션을 실행할 때, 웹 애플리케이션의 원본 폴더는 건드리지 않고, 원본을 복사한 실행용 폴더를 만들어서 사용
 - 이 기능은 파일 업로드를 할 때에는 문제가 되므로 해제
- 이클립스의 아래쪽 창에서 "Servers" 탭을 선택
- 톰캣 서버가 동작 중이라면 중지
 - "Tomcat v9.0 Server..."을 한 번만 클릭했을 때 오른쪽에 빨간색 정지 버튼이 활성화되어 있다면 톰캣이 동작 중인 것. 정지 버튼을 눌러서 톰캣 서버를 중지
- "Tomcat v9.0 Server..."을 더블클릭한다.
- 위쪽 창에 "Server Options" 파트에서 "Serve modules without publishing"을 클릭한 후 Ctrl+S로 수정 내용을 저장한다. 체크 표시가 보이는 것이 퍼블리싱을 끈 것이다.

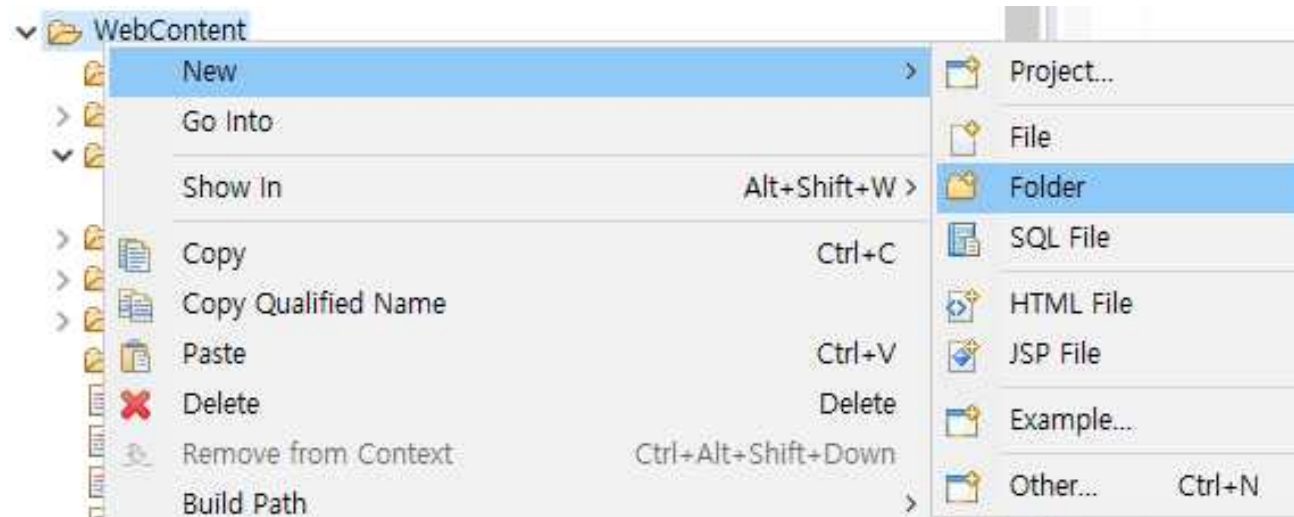
7.1 파일 업로드

- 준비 작업
 - 이클립스의 퍼블리싱 기능을 해제 (계속)



7.1 파일 업로드

- 준비 작업
 - 업로드된 파일들이 들어갈 폴더를 만들기
 - JSP 프로그램들이 있는 폴더에 업로드 파일들을 같이 저장할 수도 있지만, 그렇게 하면 나중에는 프로그램 파일과 업로드 파일이 섞여 혼란스러움
 - 이클립스의 Project Explorer에서 WebContent 폴더에 마우스 우클릭 후 New > Folder를 선택하여 "files"라는 이름으로 폴더를 만든다.



7.1 파일 업로드

■ 파일 업로드 폼

[예제 7-1] 파일 업로드를 위한 폼 (7-1.html)

```

1: <!doctype html>
2: <html>
3: <head>
4:     <meta charset="utf-8">
5: </head>
6: <body>
7:
8: <form action="7-2.jsp" method="post" enctype="multipart/form-data">
9:     업로드할 파일을 선택하세요.<br>
10:    <input type="file" name="upload"><br>
11:    <input type="submit" value="업로드">
12: </form>
13:
14: </body>
15: </html>

```

전송 방식이 POST이어야 한다.
GET 방식을 쓰면 파일 내용이 아니라 파일명만 전송된다.

enctype 속성은 폼을 전송할 때 사용할 인코딩 방법을 지정한다.
이 속성이 "multipart/form-data"가 아니면 파일명만 전송된다.

업로드할 파일을 선택하세요.

파일 선택

선택된 파일 없음

업로드

7.1 파일 업로드

- 업로드 파일 처리

[예제 7-2] 파일 업로드를 처리하는 프로그램 (7-2.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"  
2:     pageEncoding="UTF-8"%>  
3: <%@ page import="com.oreilly.servlet.MultipartRequest,  
4:     com.oreilly.servlet.multipart.DefaultFileRenamePolicy,  
5:     java.io.File" %>  
6:  
7: <!DOCTYPE html>  
8: <html>  
9: <head>  
10:     <meta charset="UTF-8">  
11: </head>  
12: <body>  
13:
```


7.1 파일 업로드

```
14: <%
15:     MultipartRequest multi = new MultipartRequest(
16:         request,                // POST 로 전달된 내용을 담은 객체
17:         application.getRealPath("/files"),    // 파일을 저장할 경로
18:         100 * 1024 * 1024,        // 최대 파일 크기 (100MB)
19:         "utf-8",                // 인코딩
20:         new DefaultFileRenamePolicy()    // 동일 파일명 처리 방법
21:     );
22:
23:     File file = multi.getFile("upload");    // 파일 객체 얻기
24:
25:     if (file == null)
26:         out.print("파일 업로드 오류!");
27:     else {
28:         out.print("File Name : " + file.getName() + "<br>");
29:         out.print("File Size : " + file.length() + "<br>");
30:     }
31: %>
32:
33: </body>
34: </html>
```

7.2 웹 하드 프로그램을 위한 테이블 생성과 메인 페이지

- 테이블 생성

[예제 7-3] 웹하드를 위한 테이블 생성 (webhard.sql)

```
1: create table webhard (  
2:     num    int          auto_increment primary key,  
3:     fname  varchar(80),  
4:     ftime  varchar(20),  
5:     fsize  int  
6: );
```

7.2 웹 하드 프로그램을 위한 테이블 생성과 메인 페이지

[예제 7-4] 웹 하드 메인 페이지 (webhard.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"  
2:     pageEncoding="UTF-8"%>  
3: <%@ page import="java.sql.*" %>  
4:  
5: <!DOCTYPE html>  
6: <html>  
7: <head>  
8:     <meta charset="UTF-8">  
9:     <style>  
10:         table { width: 700px; text-align: center; }  
11:         th    { background-color: cyan; }  
12:  
13:         .left  { text-align: left; }  
14:         .right { text-align: right; }  
15:  
16:         a:link { text-decoration: none; color: blue; }  
17:         a:hover { text-decoration: none; color: red; }  
18:     </style>  
19: </head>  
20: <body>  
21:
```

7.2 웹 하드 프로그램을 위한 테이블 생성과 메인 페이지

```

22: <form action="add_file.jsp" enctype="multipart/form-data" method="post">
23:     업로드할 파일을 선택하세요.<br>
24:     <input type="file" name="upload"><br>
25:     <input type="submit" value="업로드">
26: </form>
27: <br>
28:
29: <table>
30:     <tr>
31:         <th>파일명    </th>
32:         <th>업로드 시간</th>
33:         <th>크기      </th>
34:         <th>삭제      </th>
35:     </tr>
36:
37: <%
38:     Class.forName("org.mariadb.jdbc.Driver");
39:     try (
40:         Connection conn = DriverManager.getConnection(
41:             "jdbc:mariadb://localhost:3306/jspdb", "jsp", "1234");
42:         Statement stmt = conn.createStatement();
43:

```

7.2 웹 하드 프로그램을 위한 테이블 생성과 메인 페이지

```

44:         ResultSet rs = stmt.executeQuery("select * from webhard");
45:     ) {
46:         while (rs.next()) {
47: %>
48:         <tr>
49:             <td class="left">
50:                 <a href="files/<%=rs.getString("fname")%>">
51:                     <%=rs.getString("fname")%>
52:                 </a>
53:             </td>
54:             <td><%=rs.getString("ftime")%></td>
55:             <td class="right"><%=rs.getInt("fsize")%>&nbsp;&nbsp;&nbsp;</td>
56:             <td><a href="del_file.jsp?num=<%=rs.getInt("num")%>">X</a></td>
57:         </tr>
58: <%
59:     }
60: } catch(Exception e) {
61:     e.printStackTrace();
62: }
63: %>
64: </table>
66: </body>
67: </html>

```

업로드할 파일을 선택하세요.

파일 선택 선택된 파일 없음

업로드

파일명	업로드 시간	크기	삭제
20050709_150034.JPG	2019-06-28 13:39:52	2181008	X
20050723_121650.JPG	2019-06-28 13:40:07	2304345	X
20050722_162204.JPG	2019-06-28 13:40:12	1992603	X

7.3 파일 추가와 삭제 기능

[예제 7-5] 파일 추가 프로그램 (add_file.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"
2:     pageEncoding="UTF-8"%>
3: <%@ page import="java.sql.*" %>
4: <%@ page import="com.oreilly.servlet.MultipartRequest,
5:     com.oreilly.servlet.multipart.DefaultFileRenamePolicy,
6:     java.io.File" %>
7: <%@ page import="java.time.*" %>
8:
9: <%
10:     MultipartRequest multi = new MultipartRequest(
11:         request,
12:         application.getRealPath("/files"),    // 파일을 저장할 경로
13:         100 * 1024 * 1024,                    // 최대 파일 크기 (100MB)
14:         "utf-8",                              // 인코딩
15:         new DefaultFileRenamePolicy()         // 동일 파일명 처리 방법
16:     );
17:
18:     File file = multi.getFile("upload");       // 파일 객체 얻기
19:
```

7.3 파일 추가와 삭제 기능

```
20:     if (file != null) {
21:         Class.forName("org.mariadb.jdbc.Driver");
22:         try (
23:             Connection conn = DriverManager.getConnection(
24:                 "jdbc:mariadb://localhost:3306/jspdb", "jsp", "1234");
25:             Statement stmt = conn.createStatement();
26:         ) {
27:             // 현재 시간 얻기
28:             String curTime = LocalDate.now() + " " +
29:                 LocalTime.now().toString().substring(0, 8);
30:
31:             // 쿼리 실행
32:             stmt.executeUpdate(String.format(
33:                 "insert into webhard (fname, ftime, fsize) " +
34:                 "values ('%s', '%s', %d)",
35:                 file.getName(), curTime, (int)file.length()));
36:
37:             // 메인 페이지로 돌아가기
38:             response.sendRedirect("webhard.jsp");
39:             return;
40:         }
```

7.3 파일 추가와 삭제 기능

```
41:         } catch(Exception e) {
42:             e.printStackTrace();
43:         }
44:     }
45:
46: %>
47:
48: <!DOCTYPE html>
49: <html>
50: <head>
51:     <meta charset="UTF-8">
52: </head>
53: <body>
54:
55: <script>
56:     alert('업로드 실패 !');
57:     history.back();
58: </script>
59:
60: </body>
61: </html>
```


7.3 파일 추가와 삭제 기능

[예제 7-6] 파일 삭제 프로그램 (del_file.jsp)

```
1: <%@ page language="java" contentType="text/html; charset=UTF-8"  
2:     pageEncoding="UTF-8"%>  
3: <%@ page import="java.sql.*" %>  
4: <%@ page import="java.io.File" %>  
5:  
6: <%  
7:     int num = Integer.parseInt(request.getParameter("num"));  
8:  
9:     Class.forName("org.mariadb.jdbc.Driver");  
10:    try (  
11:        Connection conn = DriverManager.getConnection(  
12:            "jdbc:mariadb://localhost:3306/jspdb", "jsp", "1234");  
13:        Statement stmt = conn.createStatement();  
14:  
15:        // 삭제할 파일의 정보를 읽어오는 쿼리 (파일명을 알기 위해)  
16:        ResultSet rs = stmt.executeQuery(  
17:            "select * from webhard where num=" + num);  
18:    ) {
```

7.3 파일 추가와 삭제 기능

```
19:         if (rs.next()) {
20:
21:             // 지정된 파일 삭제
22:             File file = new File(application.getRealPath("/files/") +
23:                                     rs.getString("fname"));
24:             if (file != null) {
25:                 file.delete();
26:             }
27:
28:             // DB에서 파일 정보를 삭제
29:             stmt.executeUpdate(
30:                 "delete from webhard where num=" + num);
31:         }
32:
33:     } catch (Exception e) {
34:         e.printStackTrace();
35:     }
36:
37:     response.sendRedirect("webhard.jsp");
38: %>
```

연습문제

1. 웹 하드 파일 정보 리스트의 파일 크기는 현재 무조건 바이트 단위로 출력된다. 프로그램을 수정하여 다음과 같이 알아보기 쉽게 파일 크기가 출력되도록 하시오.
 - 숫자 중간(세 자리마다)에 쉼표를 넣을 것
 - 파일 크기에 따라 B, KB, MB로 단위가 바뀌어 출력되도록 할 것
 - 숫자에 천 단위마다 쉼표를 넣은 문자열을 얻으려면 다음과 같은 형식을 사용한다.
 - `String.format("%,d", 정수)`
2. 삭제 링크를 클릭하면 먼저 “정말 삭제하시겠습니까?”라고 묻고, 지우겠다고 하면 삭제하도록 수정하시오. 이를 위해서는 삭제 링크를 눌렀을 때 다음과 같은 자바스크립트 코드가 실행되도록 해야 한다.
 - `confirm("정말 삭제하시겠습니까?");`

연습문제

3. 업로드 폼에서 업로드한 사람의 이름도 입력받아 그것을 데이터베이스에 저장하고, 파일 리스트를 출력할 때도 업로더의 이름이 나오도록 수정해 보시오.
- 이를 위해서는 업로더 이름을 저장할 필드가 테이블에 있어야 한다. 이미 만들어진 테이블에 필드를 추가하는 쿼리의 형식은 다음과 같다.
 - `alter table [테이블명] add [필드명] [타입] [옵션];`
 - 따라서 다음 쿼리를 실행하면 user라는 필드를 webhard 테이블에 추가할 수 있다. 이미 만들어진 레코드의 user 필드에는 '관리자'가 들어간다.
 - `alter table webhard add user varchar(20) not null default '관리자';`
 - 한 가지 더 주의할 것은 `enctype="multipart/form-data"` 인 폼에서 입력된 내용은 `request.getParameter()`로 읽을 수 없다는 것이다. `MultipartRequest` 객체의 이름이 `multi` 라면, `multi.getParameter()`로 읽어야 한다.