

# JSP 파일 업로드하기

## 1. pom.xml 의존성 추가

```
<dependencies>
  <dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
  </dependency>
  <dependency>
    <groupId>commons-dbc</groupId>
    <artifactId>commons-dbc</artifactId>
    <version>1.4</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
  </dependency>
  <dependency>
    <groupId>commons-logging</groupId>
    <artifactId>commons-logging</artifactId>
    <version>1.2</version>
  </dependency>
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.4.6</version>
  </dependency>
  <!-- log4jdbc-remix -->
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-log4j12</artifactId>
    <version>1.7.7</version>
  </dependency>
  <dependency>
    <groupId>org.lazyluke</groupId>
    <artifactId>log4jdbc-remix</artifactId>
    <version>0.2.7</version>
  </dependency>
  <dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
  </dependency>
  <!-- fileupload -->
  <dependency>
    <groupId>commons-fileupload</groupId>
    <artifactId>commons-fileupload</artifactId>
```

```

        <version>1.3.3</version>
    </dependency>
    <!-- cos fileupload -->
    <dependency>
        <groupId>servlets.com</groupId>
        <artifactId>cos</artifactId>
        <version>05Nov2002</version>
    </dependency>
</dependencies>

```

## 2. log4j.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE log4j:configuration PUBLIC "-//APACHE//DTD LOG4J 1.2//EN" "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
    <!-- Appenders -->
    <appender name="console" class="org.apache.log4j.ConsoleAppender">
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d %5p [%c] %m%n" />
        </layout>
    </appender>
    <appender name="console-infolog" class="org.apache.log4j.ConsoleAppender">
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d %5p %m%n" />
        </layout>
    </appender>
    <!-- Application Loggers -->
    <logger name="first" additivity="false">
        <level value="debug" />
        <appender-ref ref="console" />
    </logger>
    <!-- Query Loggers -->
    <logger name="jdbc.sqlonly" additivity="false">
        <level value="INFO"/>
        <appender-ref ref="console-infolog" />
    </logger>
    <logger name="jdbc.resultsettable" additivity="false">
        <level value="INFO"/>
        <appender-ref ref="console" />
    </logger>
    <!-- Root Logger -->
    <root>
        <priority value="off"/>
        <appender-ref ref="console" />
    </root>
</log4j:configuration>

```

## 3. db.properties

```

# Maria DB 정보
# m.driver=org.mariadb.jdbc.Driver

```

```
# m.url=jdbc:mariadb://localhost:3306/jspdb
m.driver=net.sf.log4jdbc.DriverSpy
m.url=jdbc:log4jdbc:mariadb://localhost:3306/jspdb
m.username=jspuser
m.password=0000
# Oracle DB 정보
# o.driver=oracle.jdbc.driver.OracleDriver
# o.url=jdbc:oracle:thin:@127.0.0.1:1521:XE
# log4jdbc-remix
o.driver=net.sf.log4jdbc.DriverSpy
o.url=jdbc:log4jdbc:oracle:thin:@127.0.0.1:1521:XE
o.username=jspuser
o.password=0000
```

#### 4. mybatis 설정파일(MybatisConfig.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <properties resource="db.properties" />

    <typeAliases>
        <typeAlias type="java.util.HashMap" alias="hashMap"/>
    </typeAliases>

    <environments default="development">
        <environment id="development">
            <transactionManager type="JDBC" />
            <dataSource type="POOLED">
                <property name="driver" value="${o.driver}" />
                <property name="url" value="${o.url}" />
                <property name="username" value="${o.username}" />
                <property name="password" value="${o.password}" />
            </dataSource>
        </environment>
    </environments>
    <mappers>
        <mapper resource="testMapper.xml" />
    </mappers>
</configuration>
```

#### 5. 테스트 매퍼 파일(testMapper.xml)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="test">
    <select id="today" resultType="String">
        select SYSDATE from dual
    </select>
    <select id="doobae" parameterType="int" resultType="int">
        select #{num}*2 from dual
    </select>
</mapper>

```

## 6. mybatis Util 클래스(MybatisApp.java)

```

package mybatis.util;

import java.io.IOException;
import java.io.Reader;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class MybatisApp {
    private static SqlSessionFactory factory;
    static {
        String resource = "MybatisConfig.xml";
        Reader reader = null;
        try {
            reader = Resources.getResourceAsReader(resource);
            factory = new SqlSessionFactoryBuilder().build(reader);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try { reader.close(); } catch (IOException e) { ; }
        }
    }
    public static SqlSessionFactory getSessionFactory() {
        return factory;
    }
}

```

## 7. Mybatis 연결 테스트(mybatisTest.jsp)

```

<%@page import="mybatis.util.MybatisApp"%>
<%@page import="org.apache.ibatis.session.SqlSession"%>
<%@page import="org.apache.ibatis.session.SqlSessionFactoryBuilder"%>
<%@page import="org.apache.ibatis.session.SqlSessionFactory"%>
<%@page import="org.apache.ibatis.io.Resources"%>
<%@page import="java.io.Reader"%>

```

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <%
        /*
        String resource = "MybatisConfig.xml";
        Reader reader = Resources.getResourceAsReader(resource);
        SqlSessionFactory sqlMapper = new SqlSessionFactoryBuilder().build(reader);

        SqlSession sqlSession = sqlMapper.openSession();
        */
        SqlSession sqlSession = MybatisApp.getSessionFactory().openSession();
        try {

            String today = sqlSession.selectOne("test.today");
            out.println("현재 날짜 : " + today + "<br>");
            int t = sqlSession.selectOne("test.doobae",5);
            out.println("5의 두배 : " + t + "<br>");

            sqlSession.commit();
        } finally {
            sqlSession.close();
        }
    %>
</body>
</html>

```

## 9. commons-fileupload로 파일 1개 업로드하기(form1.jsp)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>파일 1개 업로드 하기</title>
</head>
<body>
    <form method="POST" enctype="multipart/form-data" action="upload1.jsp">
        이름 : <input type="text" name="name"><br/>
        설명 : <input type="text" name="note"><br/>
        파일 : <input type="file" name="upfile"><br/>
        <br/>
        <input type="submit" value="전송">
    </form>

```

```

    </form>
</body>
</html>

```

## 10. 업로드 처리 파일(upload1.jsp)

```

<%@page import="java.util.Random"%>
<%@page import="java.io.File"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%@page import="java.util.List"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<fmt:requestEncoding value="UTF-8"/>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    // commons-fileupload라이브러리를 사용하여 파일을 업로드 한다.
    // -----
    // 1. 폼전송인지 파일 전송인지를 검사해 주는 메서드(multipart/form-data인지를 판단)
    boolean isMultipart = ServletFileUpload.isMultipartContent(request);
    if(!isMultipart){
        out.println("<script>");
        out.println("alert('파일 전송폼을 이용하여 오세요!!!');");
        out.println("location.href='form1.jsp';");
        out.println("</script>");
        return;
    }

    // 2. 넘어온 내용을 파싱한다.
    // 1) DiskFileItemFactory 객체를 만든다.
    DiskFileItemFactory factory = new DiskFileItemFactory();
    // 2) ServletFileUpload 객체를 만든다.
    ServletFileUpload upload = new ServletFileUpload(factory);
    // 3) 넘어온 데이터를 파싱하여 FileItem객체 리스트를 생성한다.
    List<FileItem> items = upload.parseRequest(request);

    // 3. 데이터를 각각 처리를 해야 한다.
    String note = "", name="";
    for(FileItem item : items){ // 넘어온 데이터를 1개씩 반복 처리를 한다.
        if (item.isFormField()) {
            // 일반 폼데이터라면

            // out.println(item.getFieldName() + " : " + item.getString() + "<br>");

```

```

switch(item.getFieldName()){
case "note":
    note = item.getString("UTF-8");
    break;
case "name":
    name = item.getString("UTF-8");
    break;
}
} else {
    // 파일이라면
    String fieldName = item.getFieldName(); // 필드명 읽기
    String fileName = item.getName(); // 파일명 읽기
    String contentType = item.getContentType(); // 파일 종류
    boolean isInMemory = item.isInMemory(); // 메모리/임시폴더 저장되었는지 확인
    long sizeInBytes = item.getSize(); // 파일 크기

    // 파일을 지정 위치에 저장한다.
    String path = application.getRealPath("upload");
    // File file = new File(path + "/" + fileName);
    // 동일한 파일명을 처리하기 위해서는 저장할때 중복되지않는 이름을 만들어서 저장해야 한다.
    // 그리고 원본이름과 사본의 이름을 디비에 저장해준다.
    // 그래야 다운로드시 원본이름으로 바뀌서 다운로드하게 한다.
    // 사본 파일명을 만든다. : "시간_난수" 로 겹치지 않게 이름을 만들어 준다.
    String saveFileName = System.nanoTime() + "_" + String.format("%03d",new
Random().nextInt(100)+1);
    File file = new File(path + "/" + saveFileName);
    item.write(file);
    out.println(file.getAbsolutePath() + "에 저장 완료!!!<br>");
    out.println("필드명 : " + fieldName + "<br>");
    out.println("원본 파일 이름 : " + fileName + "<br>");
    out.println("저장 파일 이름 : " + saveFileName + "<br>");
    out.println("파일 종류 : " + contentType + "<br>");
    out.println("저장 : " + (isInMemory?"메모리":"임시폴더") + "에 저장<br>");
    out.println("파일 크기 : " + sizeInBytes + "Byte<br><hr>");
}
}
out.println("이름 : " + name + "<br>");
out.println("설명 : " + note + "<br>");
%>

</body>
</html>

```

## 11. commons-fileupload로 파일 여러개 업로드하기(form2.jsp)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">

```

```

<title>파일 여러개 업로드 하기</title>
</head>
<body>
    <form method="POST" enctype="multipart/form-data" action="upload2.jsp">
        이름 : <input type="text" name="name"><br/>
        설명 : <input type="text" name="note"><br/>
        파일 1 : <input type="file" name="upfile"><br/>
        파일 2 : <input type="file" name="upfile"><br/>
        파일 3 : <input type="file" name="upfile"><br/>
        <br/>
        <input type="submit" value="전송">
    </form>
</body>
</html>

```

## 12. 업로드 처리 파일(upload2.jsp)

```

<%@page import="java.util.Random"%>
<%@page import="java.io.File"%>
<%@page import="org.apache.commons.fileupload.FileItem"%>
<%@page import="java.util.List"%>
<%@page import="org.apache.commons.fileupload.disk.DiskFileItemFactory"%>
<%@page import="org.apache.commons.fileupload.servlet.ServletFileUpload"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<fmt:requestEncoding value="UTF-8"/>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    // commons-fileupload라이브러리를 사용하여 파일을 업로드 한다.
    // -----
    // 1. 폼전송인지 파일 전송인지를 검사해 주는 메서드(multipart/form-data인지를 판단)
    boolean isMultipart = ServletFileUpload.isMultipartContent(request);
    if(!isMultipart){
        out.println("<script>");
        out.println("alert('파일 전송폼을 이용하여 오세요!!!');");
        out.println("location.href='form2.jsp'");
        out.println("</script>");
        return;
    }

    // 2. 넘어온 내용을 파싱한다.
    // 1) DiskFileItemFactory 객체를 만든다.

    DiskFileItemFactory factory = new DiskFileItemFactory();

```



```

// 2) ServletFileUpload 객체를 만든다.
ServletFileUpload upload = new ServletFileUpload(factory);
// 3) 넘어온 데이터를 파싱하여 FileItem객체 리스트를 생성한다.
List<FileItem> items = upload.parseRequest(request);

// 3. 데이터를 각각 처리를 해야 한다.
String note = "", name="";
for(FileItem item : items){ // 넘어온 데이터를 1개씩 반복 처리를 한다.
    if (item.isFormField()) {
        // 일반 폼데이터라면
        // out.println(item.getFieldName() + " : " + item.getString() + "<br>");
        switch(item.getFieldName()){
            case "note":
                note = item.getString("UTF-8");
                break;
            case "name":
                name = item.getString("UTF-8");
                break;
        }
    } else {
        // 파일이라면
        long sizeInBytes = item.getSize(); // 파일 크기
        if(sizeInBytes>0){ // 파일 크기가 0보다 커야 파일이 올라온것이다. =====> 이부분 추가
            String fieldName = item.getFieldName(); // 필드명 읽기
            String fileName = item.getName(); // 파일명 읽기
            String contentType = item.getContentType(); // 파일 종류
            boolean isInMemory = item.isInMemory(); // 메모리/임시폴더 저장되었는지 확인

            // 파일을 지정 위치에 저장한다.
            String path = application.getRealPath("upload");
            // File file = new File(path + "/" + fileName);
            // 동일한 파일명을 처리하기 위해서는 저장할때 중복되지않는 이름을 만들어서 저장해야 한
            // 다.
            // 그리고 원본이름과 사본의 이름을 디비에 저장해준다.
            // 그래야 다운로드시 원본이름으로 바뀌서 다운로드하게 한다.
            // 사본 파일명을 만든다. : "시간_난수" 로 겹치지 않게 이름을 만들어 준다.
            String saveFileName = System.nanoTime() + "_" + String.format("%03d",new
Random().nextInt(100)+1);
            File file = new File(path + "/" + saveFileName);
            item.write(file);
            out.println(file.getAbsolutePath() + "에 저장 완료!!!<br>");
            out.println("필드명 : " + fieldName + "<br>");
            out.println("원본 파일 이름 : " + fileName + "<br>");
            out.println("저장 파일 이름 : " + saveFileName + "<br>");
            out.println("파일 종류 : " + contentType + "<br>");
            out.println("저장 : " + (isInMemory?"메모리":"임시폴더") + "에 저장<br>");
            out.println("파일 크기 : " + sizeInBytes + "Byte<br><hr>");
        }
    }
}
out.println("이름 : " + name + "<br>");
out.println("설명 : " + note + "<br>");

```

```
</body>
</html>
```

### 13. COS 라이브러리를 이용한 업로드(form3.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>파일 1개 업로드 하기</title>
</head>
<body>
    <form method="POST" enctype="multipart/form-data" action="cos_upload1.jsp">
        이름 : <input type="text" name="name"><br/>
        설명 : <input type="text" name="note"><br/>
        파일 : <input type="file" name="upfile"><br/>
        <br/>
        <input type="submit" value="전송">
    </form>
</body>
</html>
```

### 14. COS업로드 처리(cos\_upload1.jsp)

```
<%@page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Enumeration"%>
<%@page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<fmt:requestEncoding value="UTF-8"/>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    if(!request.getMethod().equals("POST")){
        out.println("<script>");
        out.println("alert('파일 전송폼을 이용하여 오세요!!!');");
        out.println("location.href='form3.jsp'");

        out.println("</script>");
    }
%>
```

```

        return;
    }
    // cos-fileupload 라이브러리를 사용하여 파일을 업로드 한다.
    String upload_path = application.getRealPath("upload"); // 업로드할 경로
    int size = 10 * 1024 * 1024; // 업로드 크기 제한
    String filename = "", name="", note="";
    int filesize = 0;
    try {
        // 이순간 모드 업로드가 이루어진다.
        MultipartRequest multi =
            new MultipartRequest(request, upload_path, size, "utf-8", new
DefaultFileRenamePolicy());

        // 폼필드의 내용을 읽자.
        name = multi.getParameter("name");
        note = multi.getParameter("note");
        out.println("이름 : " + name + "<br>");
        out.println("설명 : " + note + "<br>");
        // 파일 처리를 하자
        Enumeration<String> files = multi.getFileNames(); // 필드명 리스트
        while(files.hasMoreElements()){
            String fieldName = files.nextElement();
            String ofile = multi.getOriginalFileName(fieldName); // 원본 파일 이름
            filename = multi.getSystemName(fieldName); // 저장 파일 이름
            String contentType = multi.getContentType(fieldName); // 파일 타입
            File f1 = multi.getFile(fieldName);
            filesize = (int) f1.length(); // 파일 크기
            out.println("필드명 : " + fieldName + "<br>");
            out.println("원본 파일 이름 : " + ofile + "<br>");
            out.println("저장 파일 이름 : " + filename + "<br>");
            out.println("파일 크기 : " + filesize + "<br>");
            out.println("파일 타입 : " + contentType + "<br><hr>");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
%>
</body>
</html>

```

## 15. COS로 여러개 업로드하기(form4.jsp)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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=UTF-8">
<title>파일 여러개 업로드 하기</title>
</head>

<body>

```

```

<form method="POST" enctype="multipart/form-data" action="cos_upload2.jsp">
    이름 : <input type="text" name="name"><br/>
    설명 : <input type="text" name="note"><br/>
    <!-- 여러 파일을 업로드 하려면 반드시 name 속성의 값이 달라야 한다!!!! -->
    파일 1 : <input type="file" name="upfile1"><br/>
    파일 2 : <input type="file" name="upfile2"><br/>
    파일 3 : <input type="file" name="upfile3"><br/>
    <br/>
    <input type="submit" value="전송">
</form>
</body>
</html>

```

## 16. COS업로드 여러개 파일 처리(cos\_upload2.jsp)

```

<%@page import="java.net.URLDecoder"%>
<%@page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@page import="java.io.File"%>
<%@page import="java.util.Enumeration"%>
<%@page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions"%>
<fmt:requestEncoding value="UTF-8"/>
<!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=UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    if(!request.getMethod().equals("POST")){
        out.println("<script>");
        out.println("alert('파일 전송폼을 이용하여 오세요!!!');");
        out.println("location.href='form4.jsp';");
        out.println("</script>");
        return;
    }
    // cos-fileupload 라이브러리를 사용하여 파일을 업로드 한다.
    String upload_path = application.getRealPath("upload"); // 업로드할 경로
    int size = 10 * 1024 * 1024; // 업로드 크기 제한
    String filename = "", name="", note="";
    int filesize = 0;
    try {
        // 이순간 모드 업로드가 이루어진다.
        MultipartRequest multi =
            new MultipartRequest(request, upload_path, size, "utf-8", new
DefaultFileRenamePolicy());

```

```

// 폼필드의 내용을 읽자.
name = multi.getParameter("name");
note = multi.getParameter("note");
out.println("이름 : " + name + "<br>");
out.println("설명 : " + note + "<br>");
// 파일 처리를 하자
Enumeration<String> files = multi.getFileNames(); // 필드명 리스트
while(files.hasMoreElements()){
    String fieldName = files.nextElement();
    File f1 = multi.getFile(fieldName);
    if(f1!=null){
        filesize = (int) f1.length(); // 파일 크기
        String ofile = multi.getOriginalFileName(fieldName); // 원본 파일 이름
        filename = multi.getFilesystemName(fieldName); // 저장 파일 이름
        String contentType = multi.getContentType(fieldName); // 파일 타입
        out.println("필드명 : " + fieldName + "<br>");
        out.println("원본 파일 이름 : " + ofile + "<br>");
        out.println("저장 파일 이름 : " + filename + "<br>");
        out.println("파일 크기 : " + filesize + "<br>");
        out.println("파일 타입 : " + contentType + "<br><hr>");
        // 다운로드를 구현해보자~~~~
        // 아래와 같이 링크를 걸면 브라우저가 해석 가능한 형태의 파일이면 보여주고
        // 그렇지 않은 경우에는 다운로드가 된다.
        out.println("<a href='" + "upload/" + filename + "'" + ofile + "</a><br>");
    }
}
} catch (Exception e) {
    e.printStackTrace();
}
%>
</body>
</html>

```

## 17. 다운로드 처리하기(download.jsp)

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@ page import="java.io.*"%>
<%@ page import="java.text.*" %>
<%@ page import="java.lang.*" %>
<%@ page import="java.util.*" %>
<%@ page import="java.net.*" %>
<%

    request.setCharacterEncoding("UTF-8");
    // 파일 업로드된 경로

    String path = application.getRealPath("./upload/");

```

```

// 서버에 실제 저장된 파일명
String sf = request.getParameter("sf");
// 실제 내보낼 파일명
String of = request.getParameter("of");

InputStream in = null;
OutputStream os = null;
File file = null;
boolean skip = false; // 존재하지않는 파일일경우 패스
String client = "";
try{
    // 파일을 읽어 스트림에 담기
    try{
        file = new File(path, sf);
        in = new FileInputStream(file);
    }catch(FileNotFoundException fe){
        skip = true;
    }

    // 파일 다운로드 헤더 지정
    response.reset() ;
    response.setContentType("application/octet-stream"); // 현재 데이터가 스트림이라고 알려준
다.

    // response.setHeader("Content-Description", "JSP Generated Data");
    if(!skip){ // 파일이 존재 한다면
        // 한글 파일명 처리

        // 브라우저 종류
        client = request.getHeader("User-Agent");
        if(client.indexOf("Trident")==-1){ // IE
            of = new String(of.getBytes("utf-8"),"iso-8859-1");
        }else{ // IE ext
            of = URLEncoder.encode(of, "UTF-8" ).replaceAll("\\\\+", "%20" );
        }

        response.setHeader("Content-Disposition", "attachment; filename=\"" + of + "\"");
        response.setHeader("Content-Type", "application/octet-stream; charset=utf-8");
        response.setHeader ("Content-Length", ""+file.length() );

        // 출력스트림 얻기
        // getOutputStream() has already been called for this response - error!!!
        // JSP에서는 SERVLET으로 변환될 때 내부적으로 out 객체가 자동으로 생성하기 때문에
        // out객체를 만들면 충돌이 일어나서 저런 메시지가 뜨는 것이다.
        // 그래서 먼저 out를 초기화하고 생성하면 된다.
        out.clear();
        out = pageContext.pushBody();

        os = response.getOutputStream();
        // 복사
        byte b[] = new byte[(int)file.length()]; // 파일 크기만큼 배열선언
        int leng = 0;
        while( (leng = in.read(b)) > 0 ){ // 읽기

            os.write(b,0,leng); // 쓰기

```

```
        os.flush();
    }
    }else{
        response.setContentType("text/html;charset=UTF-8");
        out.println("<script language='javascript'>alert('파일을 찾을 수 없습니  
다');history.back();</script>");
    }
    in.close();
    os.close();
    }catch(Exception e){
        e.printStackTrace();
    }
    %>
```