



1-1. 학습 목표

파일을 전송하여 서버로 업로드 하는 자료실 형태의 폼을 학습 파일 업로드 폼, 구현, 실행

1-2. form 태그

form 태그의 enctype 속성

application/www-form-urlencoded

- 디폴트값
- 폼데이터는 서버로 전송되기 전에 URL-Encode 됨

multipart/form-data

- 파일이나 이미지를 서버로 전송할 경우 이 방식을 사용
- 만약 속성 설정을 하지 않으면 웹 서버로 데이터를 넘길때 파일의 경로명만 전송 되는 문제 발생(파일 전송 안됨)
- 메소드는 post로 지정해야 함

text/plain

• 인코딩을 하지 않은 문자 상태로 전송

1-3. input 태그

type 속성

- file 로 설정
- <input type="file" name="selectFile">

1-4. 서블릿

COS 서블릿

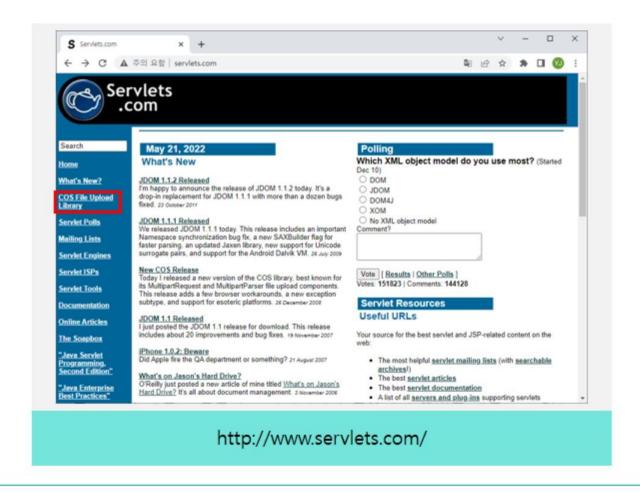
- http://www.servlets.com/
- COS(com.oreilly.servlet)
- 파일 업로드 관련

참고) 2019년도에 javaee에서 jakartaee로 변경됨

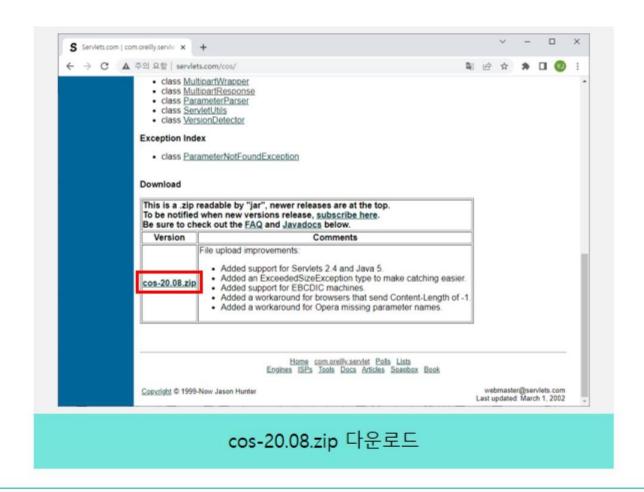
Tomcat 10은 jakartaee만 지원(Tomcat 9은 javaee 지원)

Tomcat 9은 최신 java 지원 안함

1-4. 서블릿



1-4. 서블릿



1-4. 서블릿



1-4. 서블릿

MultipartRequest 클래스

생성자 형식

 public MultipartRequest(javax.servlet.http.HttpServletRequest request, java.lang.String saveDirectory, int maxPostSize, java.lang.String encoding, FileRenamePolicy policy)

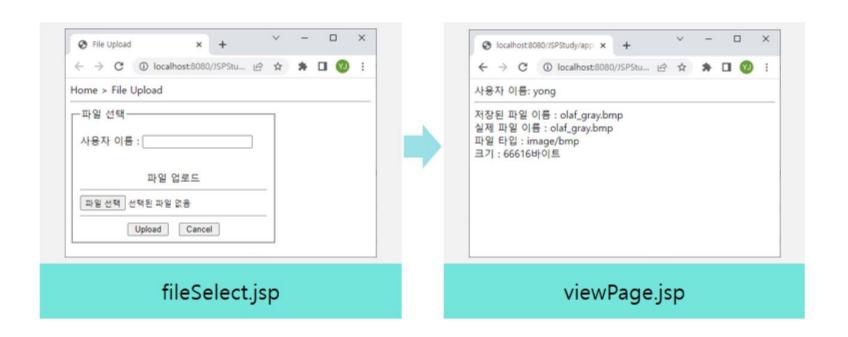
참고) javax를 jakarta로 마이그레이션 해야 사용 가능

1-4. 서블릿

MultipartRequest 클래스의 메서드

반환 자료형	메서드	기능 설명
java.lang.String	getContentType(java.lang.String name)	업로드 된 파일의 컨텐트 타입을 반환하고 업로드된 파일이 없으면 null을 반환
java.io.File	getFile(java.lang.String name)	서버 상에 업로드된 파일의 객체를 반환하고 업로드 된 파일 없으면 null을 반환
java.util.Enumeration	getFileNames()	input 태그 속성이 file로 된 파라미터의 이름들을 반환하고 업로드된 파일이 없으면 비어있는 Enumeration 객체를 반환
java.lang.String	getFilesystemName(java.lang.String name)	서버에 실제로 업로드 된 파일명을 반환
java.lang.String	getOriginalFileName(java.lang.String name)	서버에 업로드된 파일명을 반환하고 이때의 파일명은 파일 중복을 고려한 파일명 변경 전의 이름을 반환
java.lang.String	getParameter(java.lang.String name)	스트링으로 주어진 이름에 대한 값을 반환하고 값 없이 파라미터가 전송되었거나 해당되는 이름의 파라미터가 전송이 안 되었을 경우 null 값을 반환
java.util.Enumeration	getParameterNames()	모든 파라미터의 이름을 Enumeration 객체로 반환
java.lang.String[]	getParameterValues(java.lang.String name)	주어진 이름에 대한 값을 스트링 배열로 반환하고 파라미터가 전송되지 않았을 때는 null 값을 반환

1-5. 업로드 과정



• 파일 선택 페이지

• 저장된 결과 페이지

1-6. 소스코드

fileSelect.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"</p>
  pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title> File Upload </title>
</head>
<body>
 Home > File Upload
  <hr>
  <form method="post" enctype="multipart/form-data" action="viewPage.jsp">
    <fieldset style="width:300px">
      <legend> 파일 선택 </legend>
        사용자 이름:
        <input type="text" name="user"><br><br>
        <div align="center">
          파일 업로드
        </div>
        <hr>
```

1-6. 소스코드

fileSelect.jsp

```
<input type="file" name="uploadFile"><br>
        <hr>
      <div align="center">
        <input type="submit" value=" Upload "> &nbsp;
        <input type="reset" value=" Cancel ">
      </div>
    </fieldset>
  </form>
</body>
</html>
```

1-6. 소스코드

viewPage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"</p>
  pageEncoding="UTF-8"%>
<%@ page import="com.oreilly.servlet.MultipartRequest"%>
<%@ page import="com.oreilly.servlet.multipart.DefaultFileRenamePolicy"%>
<%@ page import="java.util.*"%>
<%@ page import="java.io.*"%>
<%
  String saveFolder = "파일 저장을 위한 폴더의 절대 경로";
 String encType = "UTF-8";
  int maxSize = 5 * 1024 * 1024;
  try {
    MultipartRequest multi = null;
    multi = new MultipartRequest(request, saveFolder, maxSize,
                       encType, new DefaultFileRenamePolicy());
    String user = multi.getParameter("user");
    out.println("사용자 이름: " + user + "<br>");
    out.println("<hr>");
```

1-6. 소스코드

viewPage.jsp

```
String fileName = multi.getFilesystemName("uploadFile");
    String original = multi.getOriginalFileName("uploadFile");
   String type = multi.getContentType("uploadFile");
    File f = multi.getFile("uploadFile");
    out.println("저장된 파일 이름:" + fileName + "<br>");
   out.println("실제 파일 이름:"+original+"<br>");
    out.println("파일 타입:"+type+"<br>");
    if (f != null) {
      out.println("크기:"+f.length()+"바이트");
      out.println("<br>");
  } catch (IOException ioe) {
    System.out.println(ioe);
 } catch (Exception ex) {
    System.out.println(ex);
%>
```

