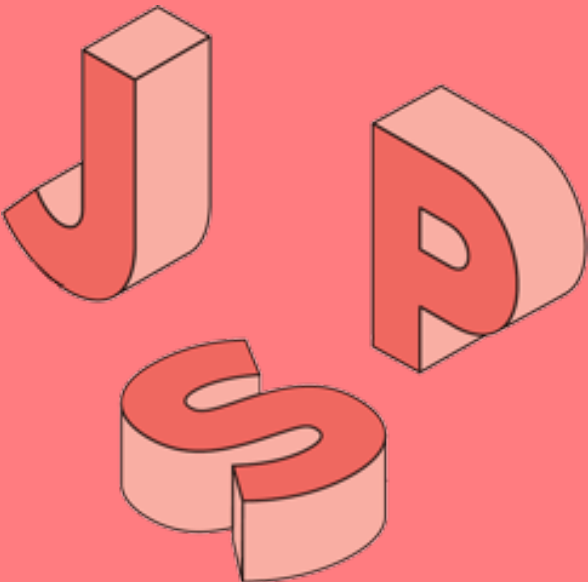


Chapter 07

JSP 오픈소스기능



Section 01

JSP에서 파일 업로드

- 15.1.1 파일 업로드 라이브러리 설치

1. jakarta.apache.org로 접속한 후 왼쪽 메뉴에서 Commons를 클릭합니다.



The Apache Jakarta Project
[http:// jakarta.apache.org/](http://jakarta.apache.org/)

Support

- [License](#)
- [Mailing Lists](#)
- [Jakarta Wiki](#)

Ex-Jakarta

- [Ant](#)
- [Avalon](#)
- [BCEL](#)
- [BSF](#)
- [Commons](#)**
- [DB](#)
- [Excalibur](#)
- [Gump](#)
- [HiveMind](#)
- [HttpComponents](#)
- [James](#)
- [JCS](#)
- [JMeter](#)

Welcome to The Apache Jakarta™ Project

Founded in 1999, the Jakarta Project housed a diverse set of Jakarta subprojects began to become full top-level Apache projects, join other TLPs (Commons), or in some cases

News

Latest Jakarta News

- 21 December 2011 - [Jakarta Retired](#)
- 26 October 2011 - [JMeter becomes a top level project](#)
- 03 October 2011 - [Apache JMeter 2.5.1 Released](#)
- 11 September 2011 - [BSF moves to Apache Commons](#)
- 17 August 2011 - [Apache JMeter 2.5 Released](#)
- 05 August 2011 - [Cactus moves to Apache Attic](#)
- 25 June 2011 - [JCS moves to Apache Commons](#)
- 25 June 2011 - [BCEL moves to Apache Commons](#)
- 17 April 2011 - [Regexr is retired](#)

JSP에서 파일 업로드

2. 페이지 왼쪽 중간쯤에 위치한 FileUpload를 클릭합니다.

DbUtils	JDBC helper library.
Digester	XML-to-Java-object mapping utility.
Email	Library for sending e-mail from Java.
Exec	API for dealing with external process execution and environment management in Java.
FileUpload	File upload capability for your servlets and web applications.
Functor	A functor is a function that can be manipulated as an object, or an object representing a single, generic function.
Geometry	Space and coordinates.
Imaging (previously called Sanselan)	A pure-Java image library.
IO	Collection of I/O utilities.

3. FileUpload 1.3.3 버전을 찾아서 [here](#)를 클릭합니다.

Downloading

Full Releases

FileUpload 1.3.3 - 13 June 2017

- Download the binary and source distributions from a mirror site [here](#)

FileUpload 1.3.2 - 26 May 2016

- Download the binary and source distributions from the archive site [here](#)

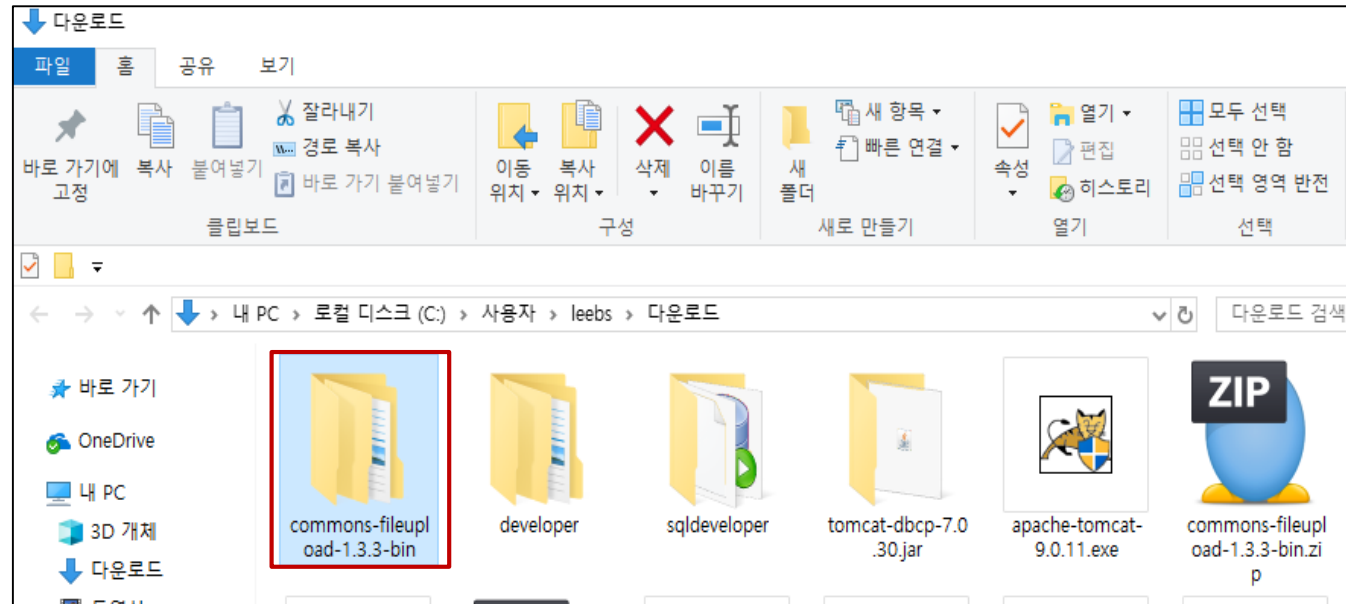
JSP에서 파일 업로드

4. commons-fileupload-1.3.3-bin.zip을 클릭해 다운로드합니다.



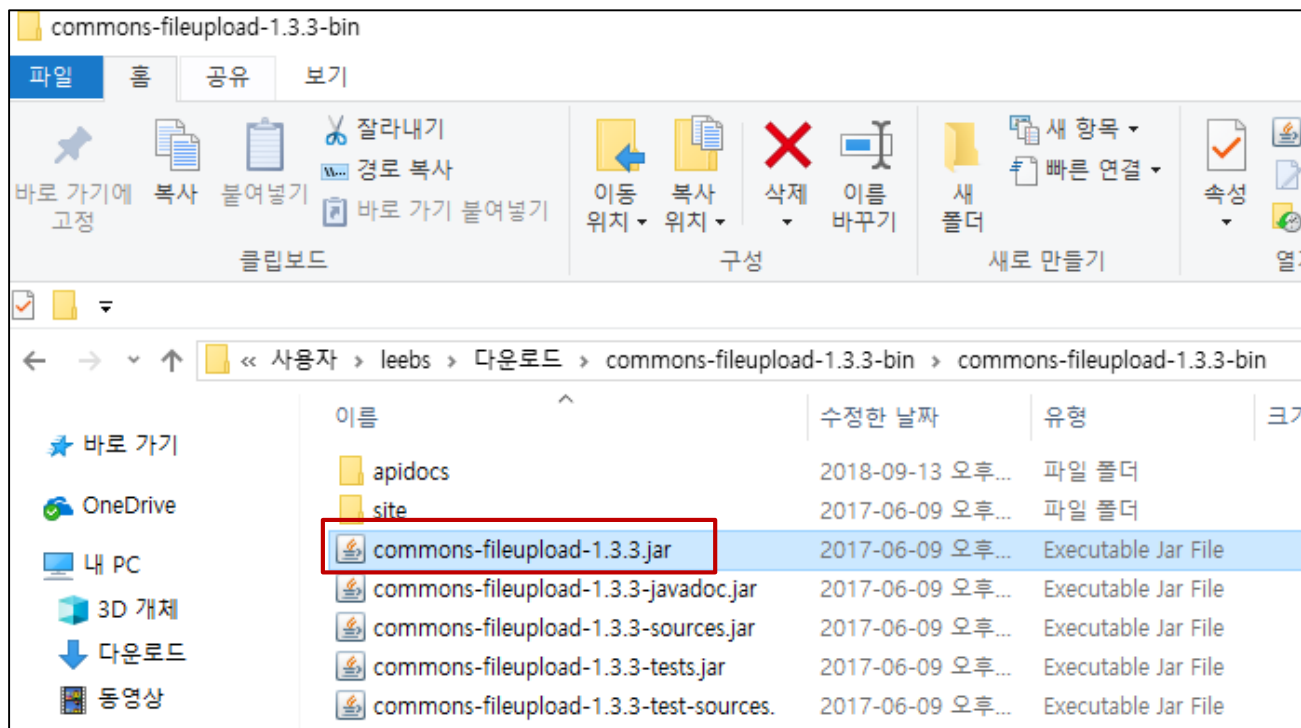
JSP에서 파일 업로드

5. zip 파일의 압축을 풉니다.



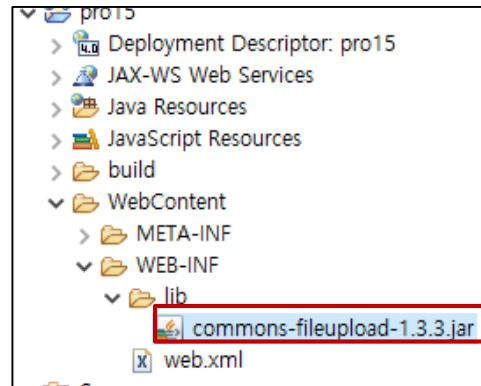
JSP에서 파일 업로드

6. 압축을 푼 폴더의 하위 폴더인 commons-fileupload-1.3.3-bin에 위치한 commons-fileupload-1.3.3.jar 파일을 복사합니다.



JSP에서 파일 업로드

7. 프로젝트 pro15의 WEB-INF 하위에 있는 lib 폴더에 붙여 넣습니다.



JSP에서 파일 업로드

- 15.1.2 commons-io-2.6.jar 파일 설치

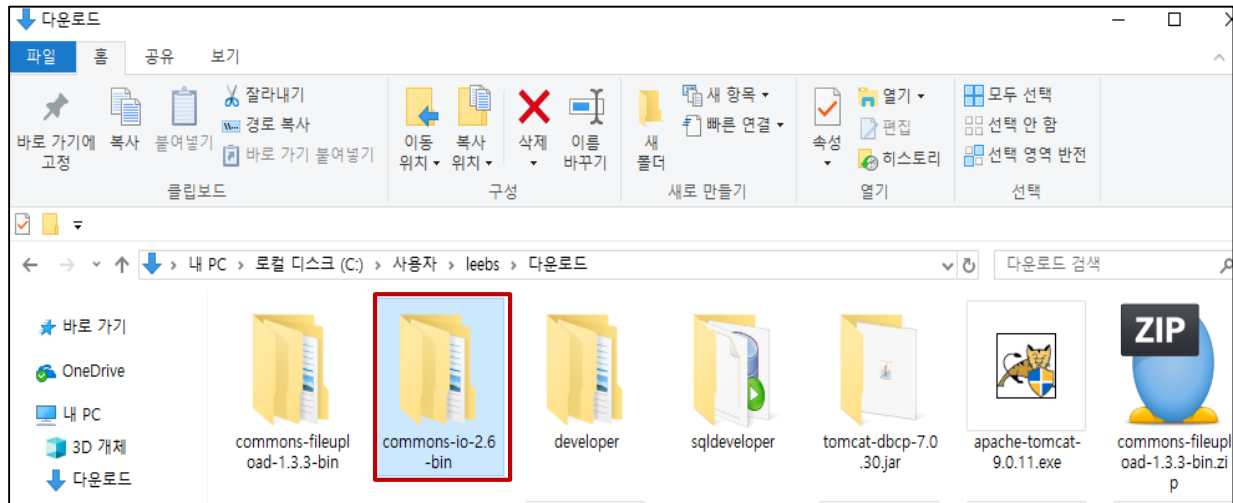
1. 다음 링크로 접속한 후 commons-io-2.6-bin.zip을 클릭해 다운로드합니다.

https://commons.apache.org/proper/commons-io/download_io.cgi



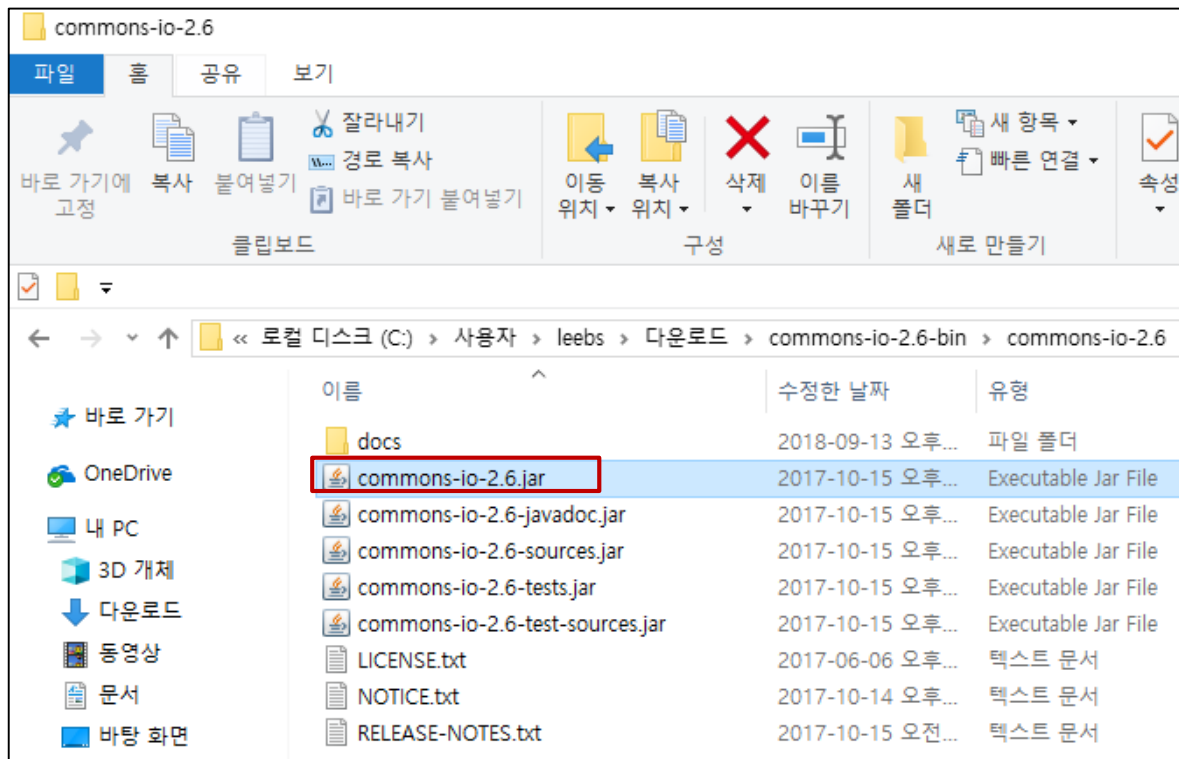
JSP에서 파일 업로드

2. 로컬 PC의 여러분이 원하는 폴더에 zip 파일의 압축을 풉니다.

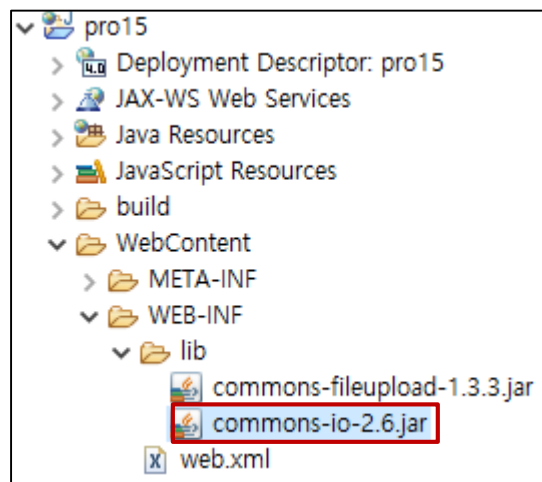


JSP에서 파일 업로드

3. commons-io-2.6-bin 폴더로 이동한 후 commons-io-2.6.jar 파일을 복사해 이클립스 프로젝트의 WEB-INF/lib 폴더에 붙여 넣습니다.



JSP에서 파일 업로드



- 15.1.3 파일 업로드 관련 API

DiskFileItemFactory 클래스가 제공하는 메서드

메서드	기능
setRepository()	파일을 저장할 디렉토리를 설정합니다.
setSizeThreshold()	최대 업로드 가능한 파일 크기를 설정합니다.

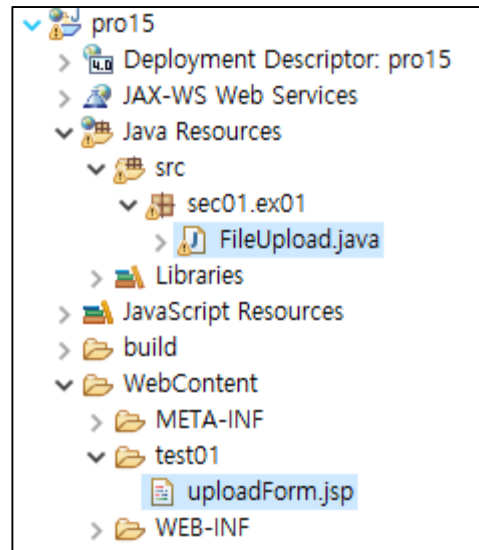
ServletFileUpload 클래스가 제공하는 메서드

메서드	기능
parseRequest()	전송된 매개변수를 List 객체로 얻습니다.
getItemIterator()	전송된 매개변수를 Iterator 타입으로 얻습니다

JSP에서 파일 업로드

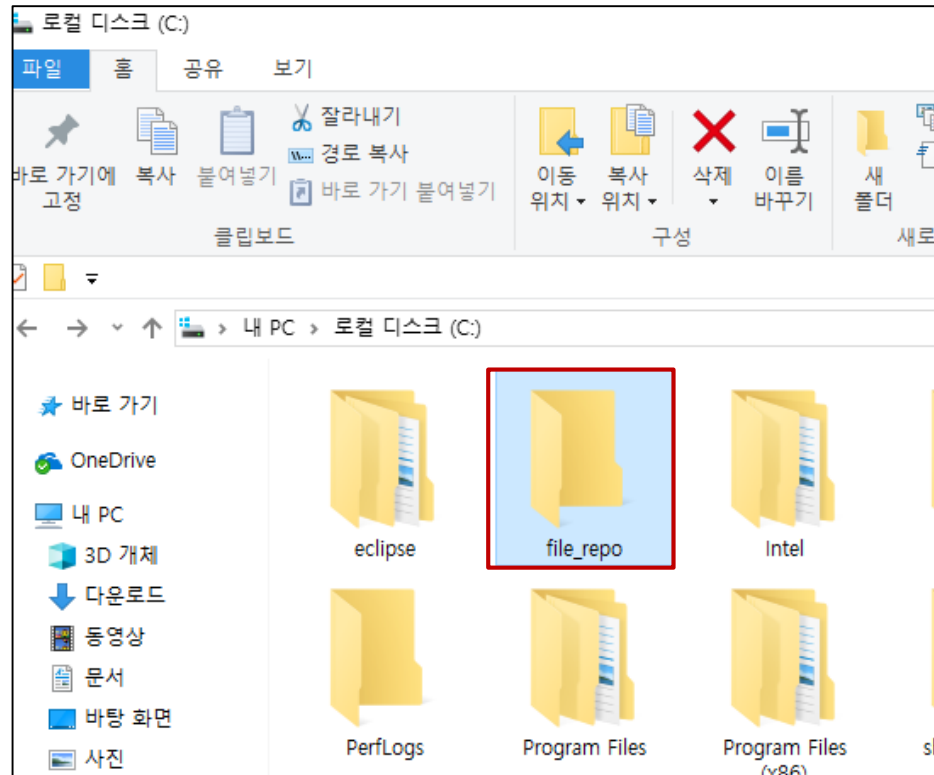
- 15.1.4 JSP 페이지에서 파일 업로드

1. sec01.ex01 패키지를 만들고 FileUpload 클래스를 생성합니다. 또 test01 폴더를 생성하고 실습 파일 uploadForm.jsp를 추가합니다.



JSP에서 파일 업로드

2. 파일을 업로드할 때 사용할 저장소를 다음과 같이 C 드라이브 아래에 만듭니다. 여기서는 폴더 이름을 file_repo로 하였습니다.



JSP에서 파일 업로드

3. uploadForm.jsp를 다음과 같이 작성합니다..

코드 15-1 pro15/WebContent/test01/uploadForm.jsp

...

<body>

<form action="\${contextPath}/upload.do"

method="post" enctype="multipart/form-data" >

파일1: <input type="file" name="file1" >

파일2: <input type="file" name="file2" >

매개변수1: <input type="text" name="param1" >

매개변수2: <input type="text" name="param2" >

매개변수3: <input type="text" name="param3" >

<input type="submit" value="업로드" >

</form>

</body>

서블릿에 요청해 파일을 업로드합니다.

파일 업로드 시 반드시 enctype을 multipart/form-data로 설정해야 합니다.

JSP에서 파일 업로드

코드 15-1 pro15/WebContent/test01/uploadForm.jsp

```
...
<body>
  <form action="${contextPath}/upload.do"
        method="post" enctype="multipart/form-data" >
    파일1: <input type="file" name="file1" ><br>
    파일2: <input type="file" name="file2" > <br>
    매개변수1: <input type="text" name="param1" > <br>
    매개변수2: <input type="text" name="param2" > <br>
    매개변수3: <input type="text" name="param3" > <br>
    <input type="submit" value="업로드" >
  </form>
</body>
```

서블릿에 요청해 프

파일 업로드 시 번
form-data로 설정

multipart/form-data

4. 파일 업로드를 처리하는 서블릿인 FileUpload 클래스를 다음과 같이 작성합니다.

코드 15-2 pro15/src/sec01/ex01/FileUpload.java

```
package sec01.ex01;

...
private public void doHandle(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    String encoding="utf-8";
    File currentDirPath =new File("C:\\file_repo");
    DiskFileItemFactory factory = new DiskFileItemFactory();
    factory.setRepository(currentDirPath );
    factory.setSizeThreshold(1024*1024);
    ServletFileUpload upload=new ServletFileUpload(factory);
    try{
        List items = upload.parseRequest(request);
        for(int i=0; i < items.size();i++) {
            FileItem fileItem = (FileItem) items.get(i);
            if(fileItem.isFormField()) {
                System.out.println(fileItem.getFieldName()+ "=" +fileItem.getString(encoding));
            }else{
                System.out.println("매개변수이름:"+fileItem.getFieldName());
                System.out.println("파일이름:"+fileItem.getName());
                System.out.println("파일크기:"+fileItem.getSize( ) + "bytes");
                if(fileItem.getSize() > 0) {
                    int idx = fileItem.getName().lastIndexOf("\\");
                    if(idx ==-1) {
                        idx = fileItem.getName().lastIndexOf("/");
                        String fileName = fileItem.getName().substring(idx+1);
                        File uploadFile = new File(currentDirPath +"\\\\"+ fileName);
                        fileItem.write(uploadFile);
                    } //end if
                } //end if
            } //end for
        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

업로드할 파일 경로를 지정합니다.

파일 경로를 설정합니다.

최대 업로드 가능한 파일 크기를 설정합니다.

request 객체에서 매개변수를 List로 가져옵니다.

파일 업로드창에서 업로드된 항목들을
하나씩 가져옵니다.

폼 필드이면 전송된 매개변수 값을
출력합니다.

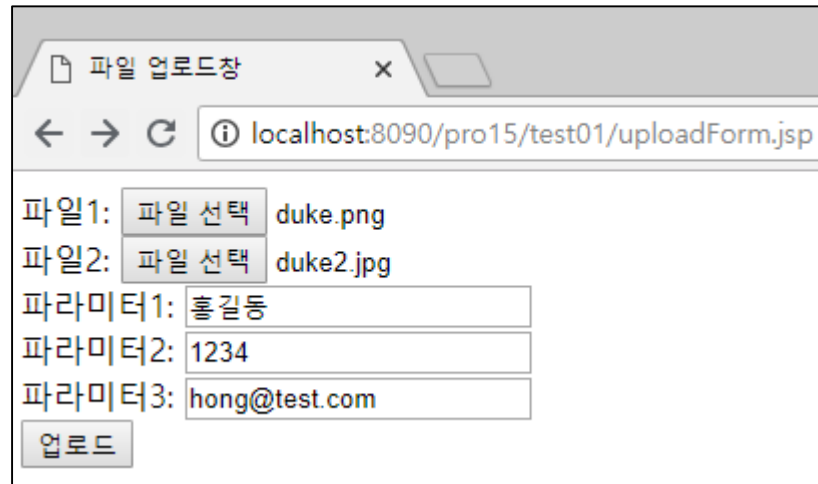
폼 필드가 아니면 파일 업로드 기능을 수행합니다.

업로드한 파일 이름을 가져옵니다.

업로드한 파일 이름으로 저장소에 파일을 업로드합니다.

JSP에서 파일 업로드

5. <http://localhost:8090/pro15/test01/uploadForm.jsp>로 요청하여 파일 업로드창을 엽니다.



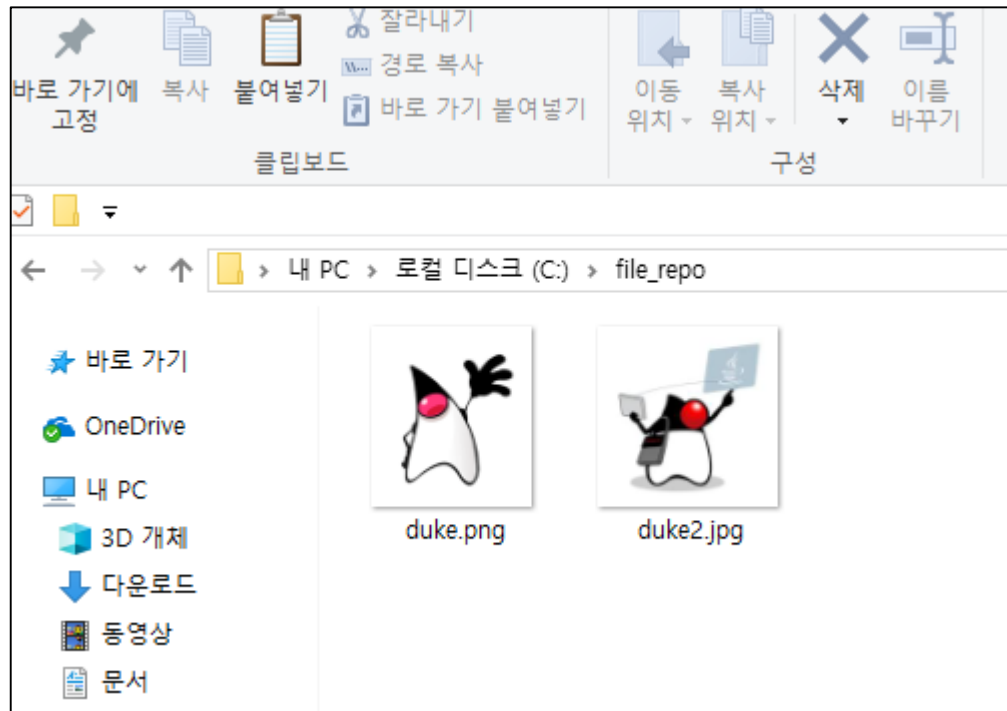
The screenshot shows a web browser window with the title '파일 업로드창' (File Upload Window). The address bar displays 'localhost:8090/pro15/test01/uploadForm.jsp'. The form contains the following fields:

- 파일1: 파일 선택 duke.png
- 파일2: 파일 선택 duke2.jpg
- 파라미터1: 홍길동
- 파라미터2: 1234
- 파라미터3: hong@test.com

At the bottom of the form is an '업로드' (Upload) button.

JSP에서 파일 업로드

6. 2번 과정에서 만든 파일 저장소(C:\file_repo)에 가면 업로드된 파일들을 볼 수 있습니다.



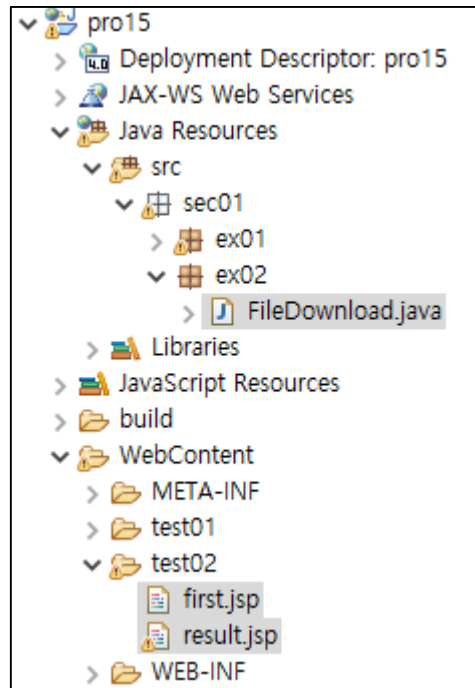
JSP에서 파일 업로드

7. 또한 이클립스의 Console 탭을 보면 업로드한 매개변수 정보와 파일 정보가 출력된 것을 확인할 수 있습니다.

```
Tomcat v9.0 Server at localhost [Apache Td  
9월 13, 2018 4:35:30 오후  
정보: Reloading Context  
파라미터명:file1  
파일명:duke.png  
파일크기:4437bytes  
파라미터명:file2  
파일명:duke2.jpg  
파일크기:4366bytes  
param1=홍길동  
param2=1234  
param3=hong@test.com
```

JSP에서 파일 업로드

1. 다음과 같이 sec01.ex02 패키지를 만들고 FileDownload 서블릿을 생성합니다. 이어서 test02 폴더를 만들고 실습 파일 first.jsp와 result.jsp를 추가합니다.



JSP에서 파일 업로드

2. 첫 번째 JSP에서 다운로드할 이미지 파일 이름을 두 번째 JSP로 전달하도록 first.jsp를 작성합니다.

코드 15-3 pro15/WebContent/test01/first.jsp

```
...
<body>
  <form method="post" action="result.jsp" >
    <input type="hidden" name="param1" value="duke.png" /> <br>
    <input type="hidden" name="param2" value="duke2.jpg" /> <br>
    <input type="submit" value="이미지 다운로드">
  </form>
</body>
```

다운로드할 파일 이름을 매개변수로 전달합니다
(duke.png나 duke2.png가 아닌 다른 파일을 업로드
했다면 해당 파일 이름으로 수정하세요).

JSP에서 파일 업로드

3. 두 번째 JSP인 result.jsp를 다음과 같이 작성합니다.

코드 15-4 pro15/WebContent/test01/result.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    isELIgnored="false"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="contextPath" value="${pageContext.request.contextPath}" />
<%
    request.setCharacterEncoding("utf-8");
%>
<html>
<head>
    <meta charset="UTF-8">
    <c:set var="file1" value="${param.param1}" />
    <c:set var="file2" value="${param.param2}" />
```

다운로드할 파일 이름을 가져옵니다.

```
    <title>이미지 파일 출력하기</title>
</head>
<body>
    매개변수 1 :
    <c:out value="${file1}" /><br>
    매개변수 2 :
    <c:out value="${file2}" /><br>
```

```
    <c:if test="${not empty file1}">
```

```
        <br>
```

```
    </c:if>
```

```
    <br>
```

```
    <c:if test="${not empty file2}">
```

```
        <br>
```

```
    </c:if>
```

```
    파일 내려받기 :<br>
```

```
    <a href="${contextPath}/download.do?fileName=${file2}">
```

```
        파일 내려받기 </a><br>
```

```
</body>
```

```
</html>
```

파일 이름으로 서블릿에서 이미지를 다운로드해 표시합니다.

파일 이름으로 서블릿에서 이미지를 다운로드해 표시합니다.

이미지를 파일로 다운로드합니다.

JSP에서 파일 업로드

4. 파일 다운로드 기능을 할 서블릿인 FileDownload 클래스를 다음과 같이 작성합니다.

코드 15-5 pro15/src/sec01/ex01/FileDownload.java

```
package sec02.ex01;

...

private protected void doHandle(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException{
    request.setCharacterEncoding("utf-8");
    response.setContentType("text/html; charset=utf-8");
    String file_repo="C:\\file_repo";
    String fileName = (String)request.getParameter("fileName");
    System.out.println("fileName="+fileName);
    OutputStream out = response.getOutputStream();
    String downFile=file_repo+"\\ "+fileName;
    File f=new File(downFile);
    response.setHeader("Cache-Control","no-cache");
    response.addHeader("Content-disposition", "attachment; fileName="+fileName);

    FileInputStream in=new FileInputStream(f);
    byte[] buffer=new byte[1024*8];
    while(true) {
        int count=in.read(buffer);
        if(count== -1)
            break;
        out.write(buffer,0,count);
    }
    in.close();
    out.close();
}
}
```

매개변수로 전송된 파일 이름을 읽어옵니다.

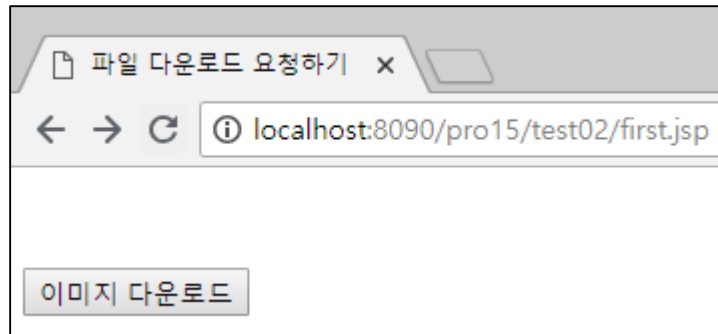
response에서 OutputStream 객체를 가져옵니다.

파일을 다운로드할 수 있습니다.

버퍼 기능을 이용해 파일에서 버퍼로 데이터를 읽어와 한꺼번에 출력합니다.

JSP에서 파일 업로드

5. <http://localhost:8090/pro15/test02/first.jsp>로 요청한 후 이미지 다운로드를 클릭합니다.



JSP에서 파일 업로드

6. 업로드한 이미지가 브라우저에 출력되면 **파일 내려받기**를 클릭해 로컬 PC에 파일을 저장합니다.

