

9

파일 업로드



이 장에서 다룰 내용

1

파일 업로드의 원리

2

가장 널리 쓰이는 업로드 모듈 COS 라이브러리

3

MultipartRequest를 이용한 파일 업로드 구현



파일 업로드의 원리

□ 파일 업로드의 원리

```
<form method="post" enctype="multipart/form-data">  
    <input type="file" name="filename">  
</form>
```

- **enctype 속성을 multipart/form-data로 설정**



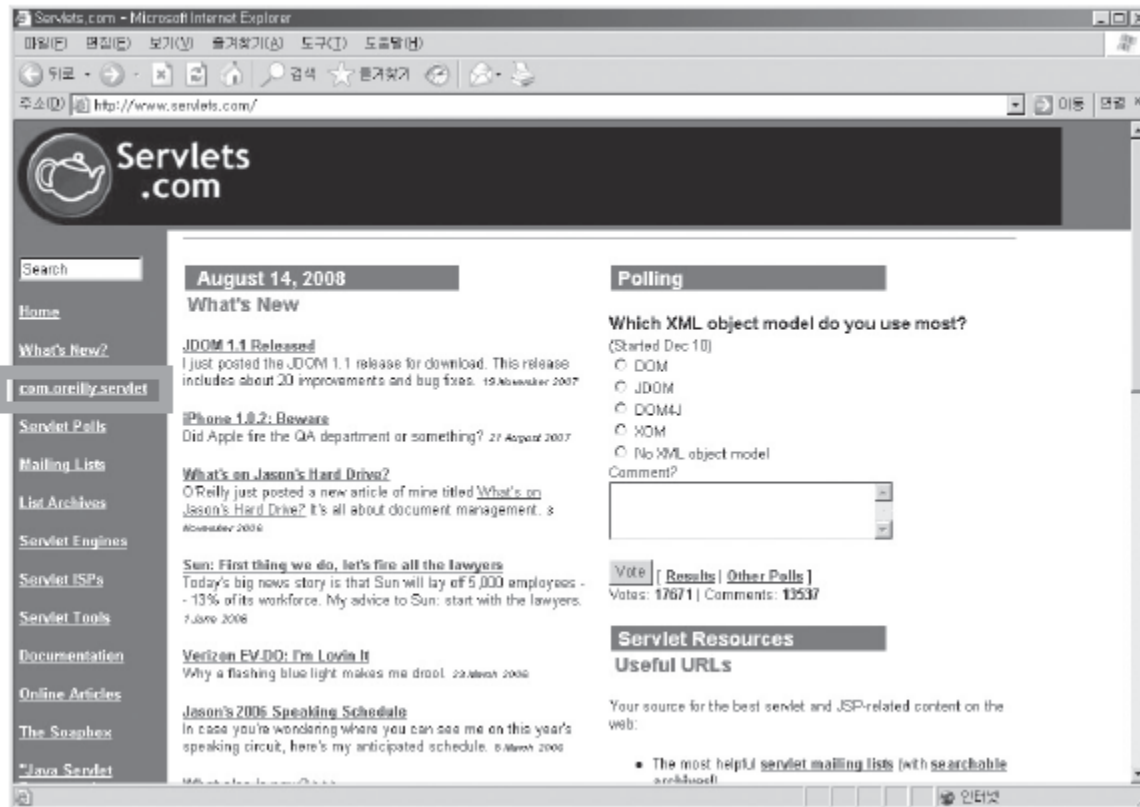
가장 널리 쓰이는 업로드 모듈 COS 라이브러리

□ COS 라이브러리 소개

- 현재 자바에서 가장 널리 쓰여지는 업로드 컴포넌트

□ COS 라이브러리 다운로드 및 설치하기

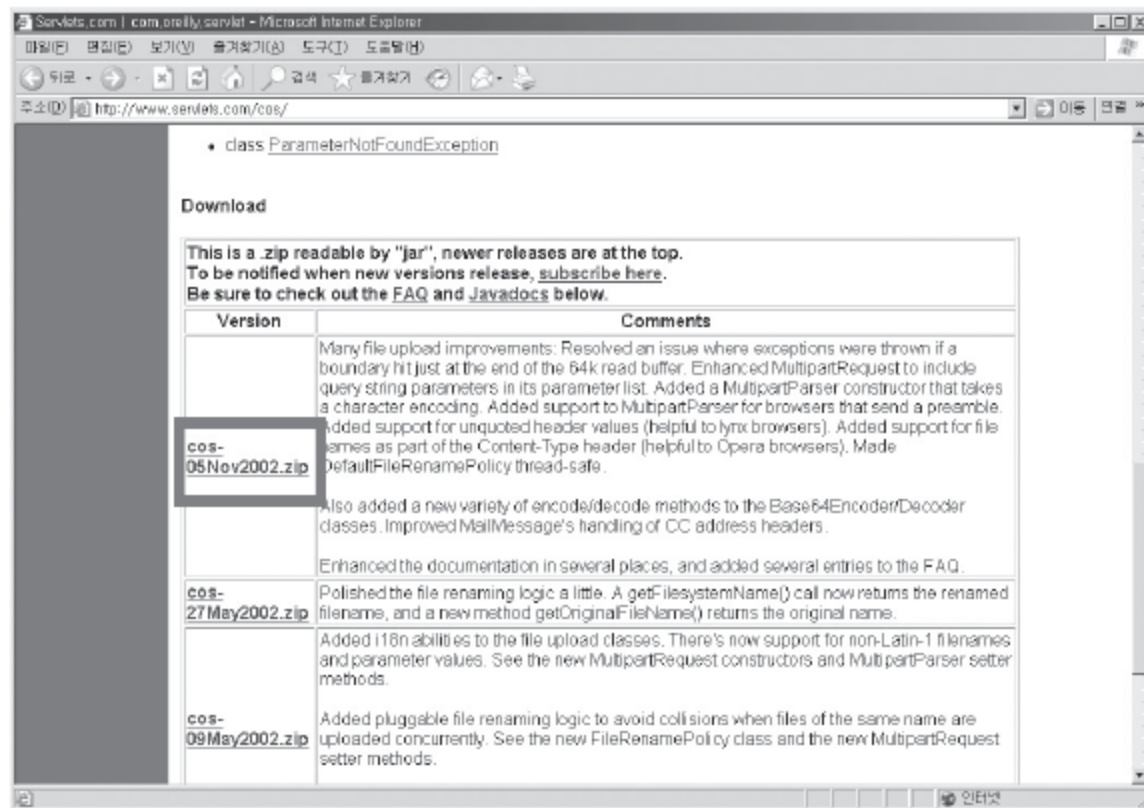
- COS 라이브러리 파일은 <http://www.servlets.com>에서 다운로드



1. <http://www.servlets.com>에 접속하고 좌측 메뉴의 [com.oreilly.servlet](#)을 클릭



가장 널리 쓰이는 업로드 모듈 COS 라이브러리

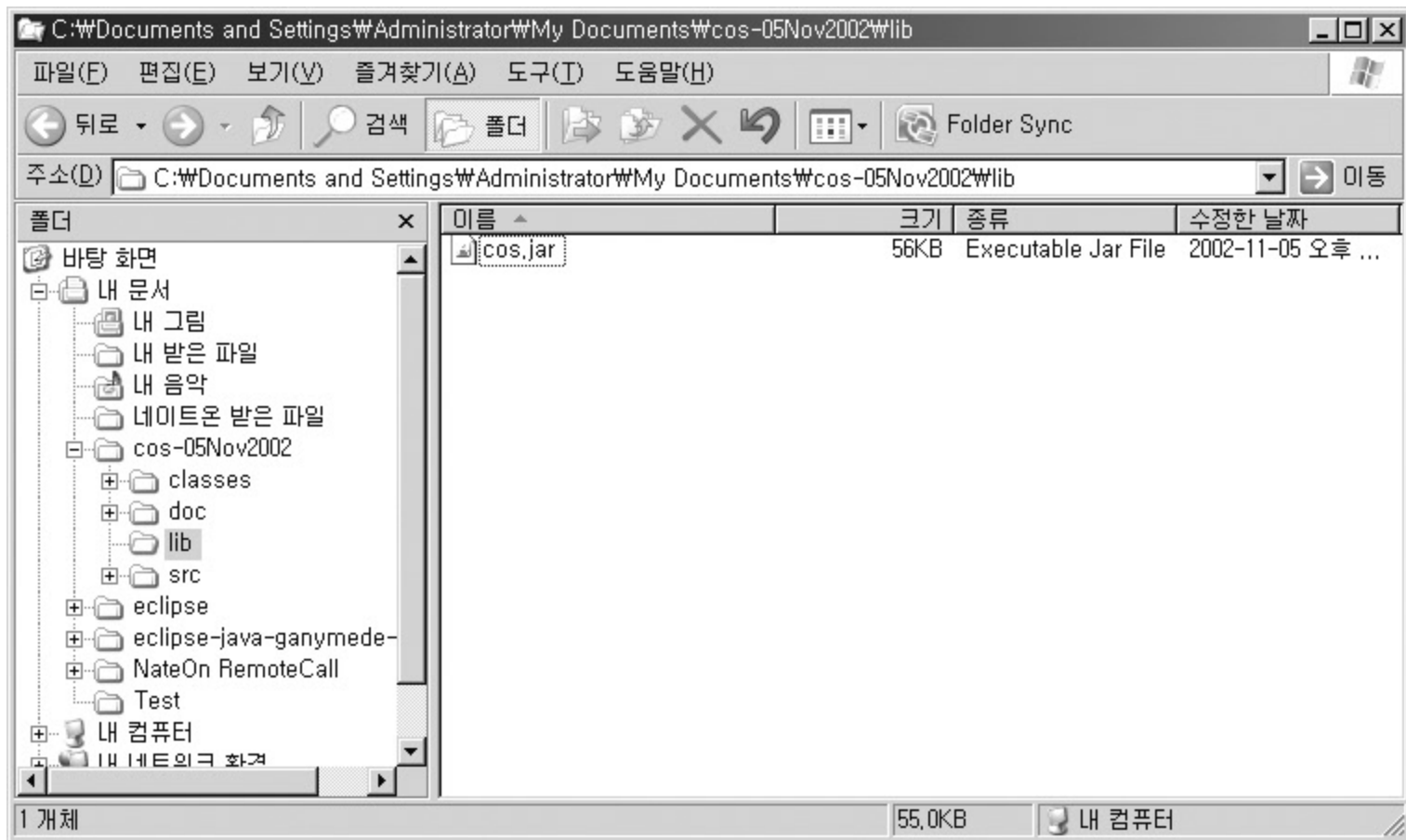


2. Cos-O5Nov2002.zip
파일을 클릭하여 다운로드



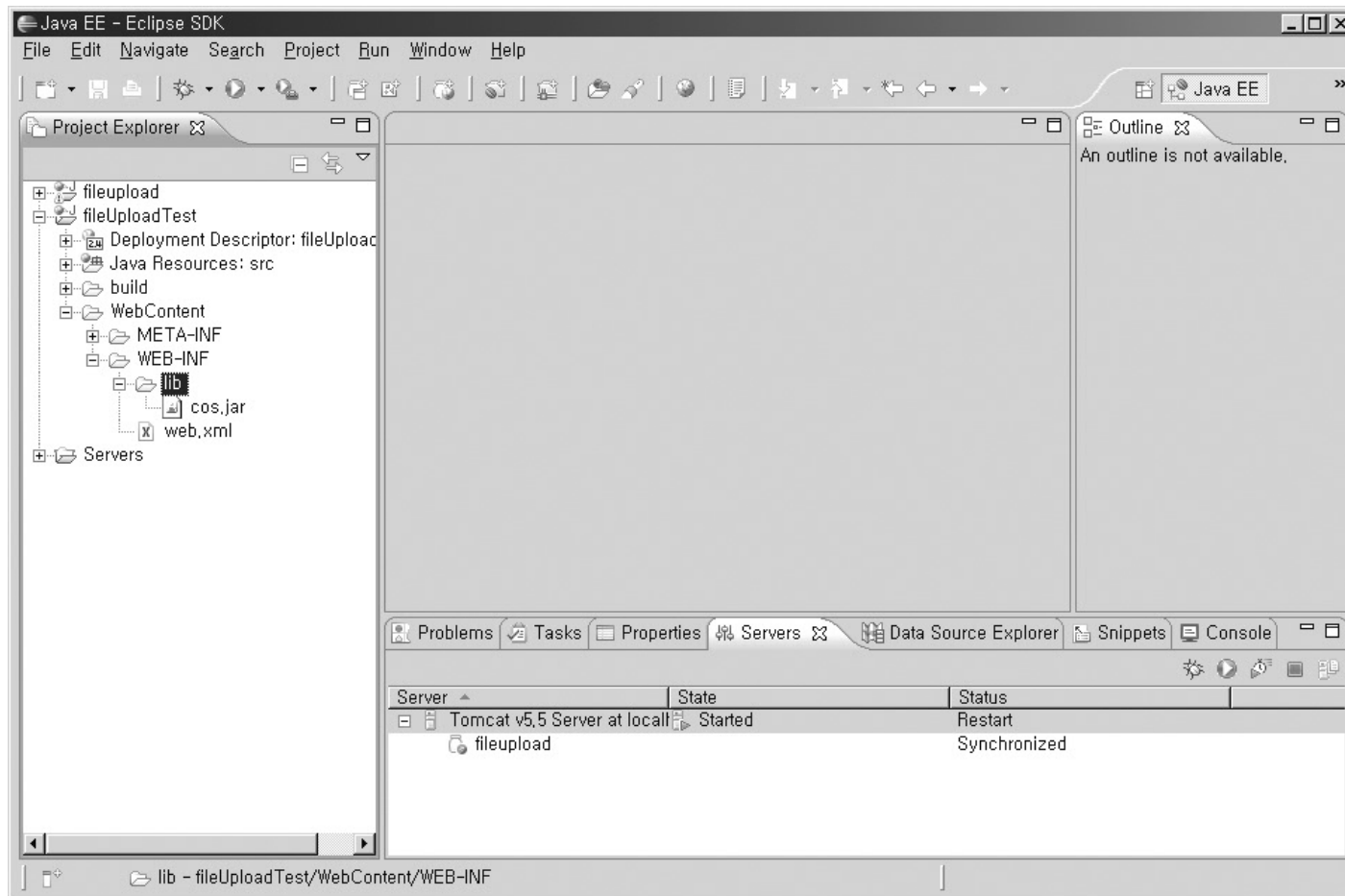
가장 널리 쓰이는 업로드 모듈 COS 라이브러리

3. 다운로드 받은 파일을 압축을 푼 후에 lib 폴더로 열어 cos.jar 파일 확인



가장 널리 쓰이는 업로드 모듈 COS 라이브러리

- 이 라이브러리를 사용하기 위해 이클립스의 라이브러리 폴더에 추가하도록 한다. Dynamic WebProject를 하나 생성하고 WebContent\WEB-INF\lib 폴더에 cos.jar를 복사한다.



가장 널리 쓰이는 업로드 모듈 COS 라이브러리

□ MultipartRequest의 특징

- COS 라이브러리에서 가장 핵심적인 역할을 하는 클래스
- 파일 업로드를 직접적으로 담당

□ MultipartRequest의 생성자

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

인자	설명
request	MultipartRequest와 연결될 request 객체를 의미한다.
saveDirectory	서버 측에 파일이 실질적으로 저장될 경로를 의미한다.
maxPostSize	한번에 업로드 할 수 있는 최대 파일 크기를 의미한다.
encoding	파일의 인코딩 방식을 의미한다.
policy	파일 이름 중복 처리를 위한 인자를 의미한다.



가장 널리 쓰이는 업로드 모듈 COS 라이브러리

□ MultipartRequest의 메소드

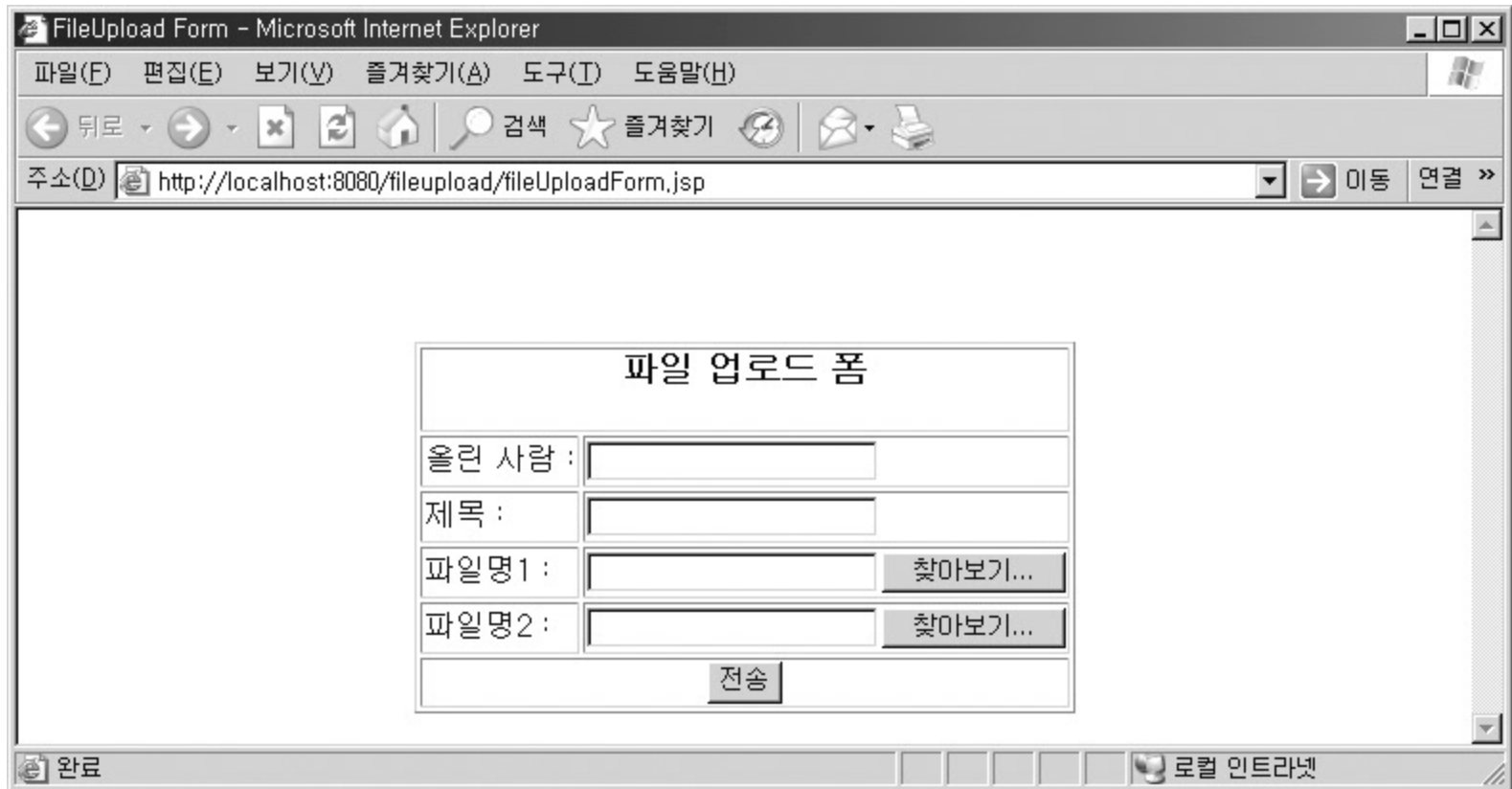
메소드	설명
getParameterNames()	폼에서 전송된 파라미터의 타입이 file이 아닌 이름들을 Enumeration 타입으로 리턴한다.
getParameterValues()	폼에서 전송된 파라미터 값들을 배열로 받아 온다.
getParameter()	Request 객체에 있는 해당 파라미터의 값을 가져온다.
getFileNames()	파일을 여러 개 업로드 할 경우 타입이 file인 파라미터 이름들을 Enumeration 타입으로 리턴한다.
getFileSystemName()	서버에 실제로 업로드 된 파일의 이름을 의미한다.
getOriginalFileName()	클라이언트가 업로드 한 파일의 원본 이름을 의미한다.
getContentType()	업로드 된 파일의 콘텐츠 타입을 얻을 때 사용한다.
getFile()	서버에 업로드 된 파일 객체 자체를 반환하는 메소드이다.



MultipartRequest를 이용한 파일 업로드 구현

□ 파일 업로드 폼 작성하기

- fileUploadForm.jsp (교재 220p 참조)



The screenshot shows a Microsoft Internet Explorer window titled "FileUpload Form - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/fileupload/fileUploadForm.jsp". The main content area contains a form titled "파일 업로드 폼" (File Upload Form). The form includes the following fields and buttons:

파일 업로드 폼	
올린 사람 :	<input type="text"/>
제목 :	<input type="text"/>
파일명1 :	<input type="text"/> <input data-bbox="1122 996 1309 1035" type="button" value="찾아보기..."/>
파일명2 :	<input type="text"/> <input data-bbox="1122 1053 1309 1092" type="button" value="찾아보기..."/>
<input data-bbox="942 1110 1014 1149" type="button" value="전송"/>	

The status bar at the bottom shows "완료" (Completed) and "로컬 인트라넷" (Local intranet).

MultipartRequest를 이용한 파일 업로드 구현

□ 업로드 페이지 작성하기

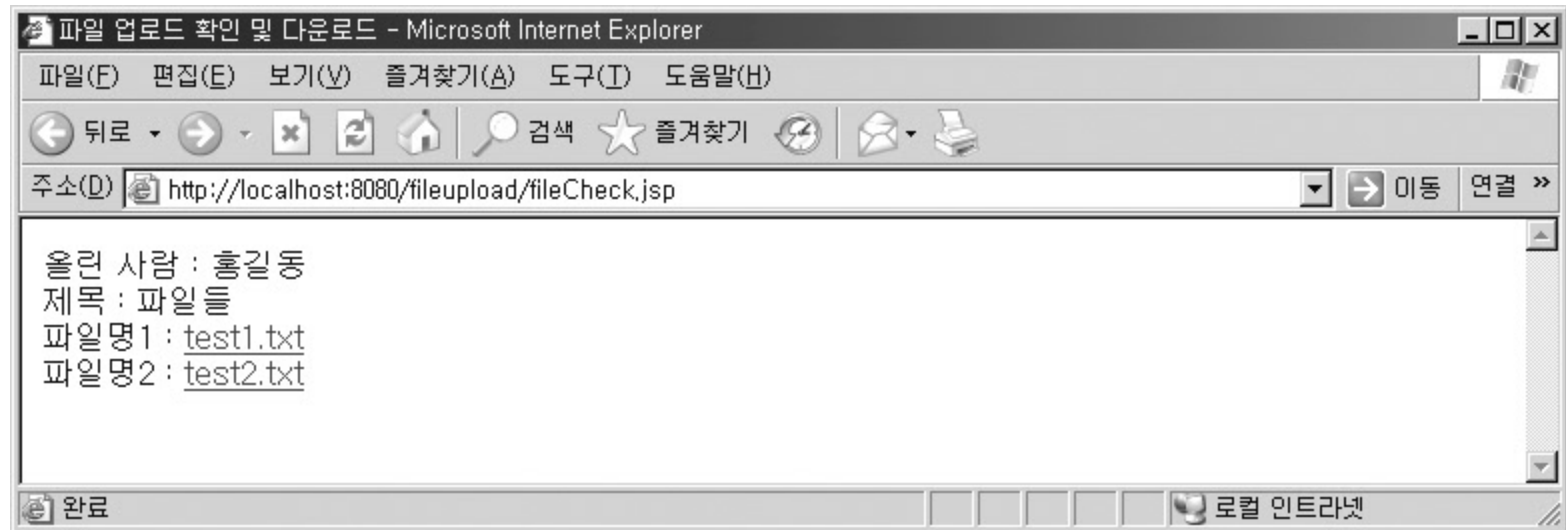
- fileUpload.jsp (교재 221p 참조)



MultipartRequest를 이용한 파일 업로드 구현

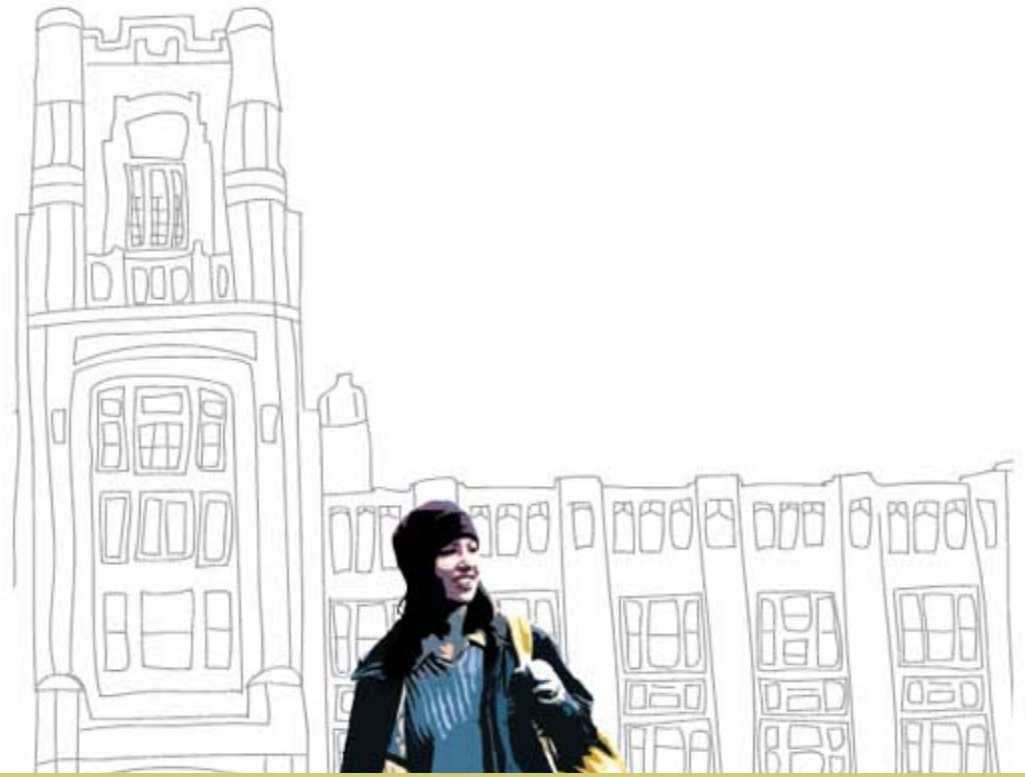
□ 업로드의 확인 및 다운로드 페이지 작성하기

- fileCheck.jsp (교재 224p 참조)



- 브라우저에서 실행되는 파일에 대해서도 다운로드 박스가 출력되게 처리하는 페이지 코드 작성
 - file_down.jsp (교재 225p 참조)





Thank You