

# 7 파일 업로드

1

파일 업로드의 개요

2

MultipartRequest를 이용한 파일 업로드

3

Common-FileUpload를 이용한 파일 업로드

4

[웹 쇼핑몰] 상품 이미지 등록하기

#### 학습목표

- 파일 업로드의 개념과 특징을 이해합니다.
- MultipartRequest를 이용한 파일 업로드 방법을 익힙니다.
- Commons-FileUpload를 이용한 파일 업로드 방법을 익힙니다.
- 파일 업로드 방법을 이용하여 웹 쇼핑몰의 상품 이미지를 등록합니다.

# 1. 파일 업로드의 개요

## ❖ 파일 업로드(file upload)

- 웹 브라우저에서 서버로 파일을 전송하여 서버에 저장하는 것
- 서버로 업로드할 수 있는 파일
  - 텍스트 파일, 바이너리 파일, 이미지 파일, 문서 등 다양한 유형이 있음
- 웹 브라우저에서 서버로 파일을 전송하기 위해 JSP 페이지에 폼 태그 사용
- 전송된 파일을 서버에 저장하기 위해 오픈 라이브러리 이용

# 1. 파일 업로드의 개요

## ❖ 파일 업로드를 위한 JSP 페이지

- 웹 브라우저에서 서버로 파일을 전송하기 위해 JSP 페이지에 폼 태그를 작성할 때 몇 가지 중요한 규칙

```
<form action = "JSP 파일" method="POST" enctype="multipart/form-data">  
  <input type="file" name="요청 파라미터 이름">  
</form>
```

- form 태그의 method 속성은 반드시 POST 방식으로 설정
- form 태그의 enctype 속성은 반드시 multipart/form-data로 설정
- form 태그의 action 속성은 파일 업로드를 처리할 JSP 파일로 설정
- 파일 업로드를 위해 input 태그의 type 속성을 file로 설정
  - 만약 여러 파일을 업로드하려면 2개 이상의 input 태그를 사용하고 name 속성에 서로 다른 값을 설정

# 1. 파일 업로드의 개요

[파일 업로드를 위한 폼 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>File Upload</title>
</head>
<body>
    <form method="post" enctype="multipart/form-data" action="process.jsp">
        <p>제목 : <input type="text" name="title">
        <p>파일 : <input type="file" name="fileName">
        <p><input type="submit" value="submit">
    </form>
</body>
</html>
```

제목 :

파일 :  [찾아보기...](#)

1

2

3

4

# 1. 파일 업로드의 개요

## ❖ 파일 업로드 처리 방법

- 단순한 자바 코드로 작성하여 처리할 수 없어 오픈 라이브러리인 cos.jar나 commonsfileupload.jar를 사용해야 함

## ❖ 파일 업로드 처리 방법

종류	특징	필요한 라이브러리
MultipartRequest 이용하기	가장 간단한 방법	cos.jar
아파치 API 이용하기	편리하고 강력한 API 제공	commons-fileupload.jar commons-io.jar

## 2. MultipartRequest를 이용한 파일 업로드

### ❖ MultipartRequest

- 웹 페이지에서 서버로 업로드되는 파일 자체만 다루는 클래스
- 웹 브라우저가 전송한 multipart/form-data 유형과 POST 방식의 요청 파라미터 등을 분석한 후 일반 데이터와 파일 데이터를 구분하여 파일 데이터에 접근
- 한글 인코딩 값을 얻기 쉽고, 서버의 파일 저장 폴더에 동일한 파일명이 있으면 파일명을 자동으로 변경
- 오픈 라이브러리 cos.jar를 배포 사이트에서 직접 다운로드해서 사용
  - 배포 사이트: <http://www.servlets.com/cos/>
  - 다운로드 파일: cos-26Dec2008.zip
  - JSP 페이지에 page 디렉티브 태그의 import 속성을 사용하여 패키지 com.oreilly.servlet.\*을 설정

#### **NOTE\_** 오픈 라이브러리 cos.jar 설치 방법

- ❶ 배포 사이트에서 cos-26Dec2008.zip 파일을 다운로드합니다.
- ❷ cos-26Dec2008.zip 파일의 압축을 풉니다.
- ❸ 압축을 푼 파일의 lib 폴더에 있는 cos.jar 파일을 찾습니다.
- ❹ 웹 애플리케이션의 /WebContent/WEB-INF/lib/ 폴더에 cos.jar 파일을 포함합니다. 또는 서버 톰캣의 /common/lib/ 폴더와 자바 JDK의 lib 폴더에 cos.jar 파일을 포함합니다.

## 2. MultipartRequest를 이용한 파일 업로드

### ❖ MultipartRequest 클래스 생성

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

### ❖ MultipartRequest 생성자의 매개변수

매개변수	설명
request	Request 내장 객체를 설정합니다.
saveDirectory	서버의 파일 저장 경로를 설정합니다.
maxPostSize	파일의 최대 크기(바이트 단위)를 설정합니다. 최대 크기를 초과하면 IOException이 발생합니다.
encoding	인코딩 유형을 설정합니다.
policy	파일명 변경 정책을 설정합니다. saveDirectory에 파일명이 중복되는 경우 덮어쓰기 여부를 설정하며, 설정하지 않으면 덮어씁니다.



## 2. MultipartRequest를 이용한 파일 업로드

[MultipartRequest 클래스 생성 예]

```
<%@ page import="com.oreilly.servlet.*"%>
<%@ page import="com.oreilly.servlet.multipart.*"%>
...(생략)...

MultipartRequest multi = new MultipartRequest(request,
    "C:\\upload", 5*1024*1024, "utf-8",
    new DefaultFileRenamePolicy())
```

## 2. MultipartRequest를 이용한 파일 업로드

### ❖ MultipartRequest 메소드

- 웹 브라우저에서 전송되는 요청 파라미터 중
  - 일반 데이터는 `getParameter( )` 메소드로 값을 받음
  - 파일의 경우 `getFileNames( )` 메소드를 이용하여 데이터를 받음

## 2. MultipartRequest를 이용한 파일 업로드

### ❖ MultipartRequest 메소드의 종류

메소드	유형	설명
getContentType(String name)	String	업로드된 파일의 콘텐츠 유형을 반환합니다. 업로드된 파일이 없으면 null을 반환합니다.
getParameter(String name)	String	요청 파라미터 이름이 name인 값을 전달받습니다.
getParameterNames()	java.util Enumeration	요청 파라미터 이름을 Enumeration 객체 타입으로 반환합니다.
getFile(String name)	java.io.File	서버에 업로드된 파일에 대한 파일 객체를 반환합니다. 업로드된 파일이 없으면 null을 반환합니다.
getFileNames()	java.util Enumeration	폼 페이지에 input 태그 내 type 속성 값이 file로 설정된 요청 파라미터의 이름을 반환합니다.
getFileSystemName(String name)	String	사용자가 설정하여 서버에 실제로 업로드된 파일명을 반환합니다. 파일명이 중복되면 변경된 파일명을 반환합니다.
getOriginalFileName(String name)	String	사용자가 업로드한 실제 파일명을 반환합니다. 파일명이 중복되면 변경 전의 파일명을 반환합니다.

## 2. MultipartRequest를 이용한 파일 업로드

[MultipartRequest 클래스의 메소드 사용 예: 요청 파라미터 정보 출력하기]

```
<%@ page contentType="text/html;charset=utf-8"%>
<%@ page import="com.oreilly.servlet.*"%>
<%@ page import="com.oreilly.servlet.multipart.*"%>
<%@ page import="java.util.*"%>
<%@ page import="java.io.*"%>
<%

    MultipartRequest multi = new MultipartRequest(request, "C:\\upload",
5 * 1024 * 1024, "utf-8",      new DefaultFileRenamePolicy());

    String title = multi.getParameter("title");
    out.println("<h3>" + title + "</h3>");

    Enumeration files = multi.getFileNames();
    String name = (String) files.nextElement();

    String filename = multi.getFilesystemName(name);
    String original = multi.getOriginalFileName(name);

    out.println("실제 파일 이름 : " + original + "<br>");
    out.println("저장 파일 이름 : " + filename + "<br>");

%>
```

실행 결과



(a) 파일명이 중복되지 않는 경우

파일 업로드  
실제 파일 이름 : Penguins.jpg  
저장 파일 이름 : Penguins.jpg

(b) 파일명이 중복되는 경우

파일 업로드  
실제 파일 이름 : Penguins.jpg  
저장 파일 이름 : Penguins1.jpg

## 2. MultipartRequest를 이용한 파일 업로드

### 예제 7-1 MultipartRequest 클래스를 이용하여 파일 업로드 및 정보 출력하기

- 오픈 라이브러리 cos.jar 파일을 다운로드하여 /WebContent/WEB-INF/lib/ 폴더에 추가

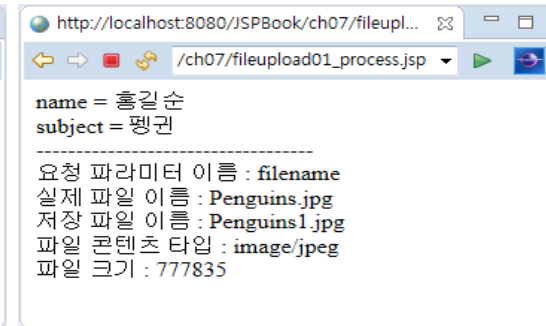
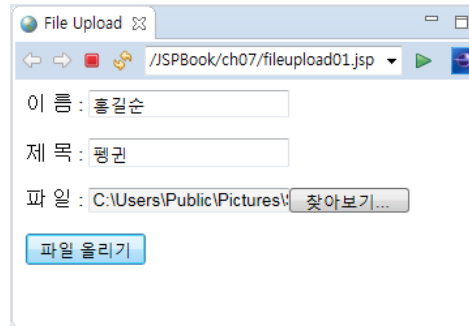
JSPBook/WebContent/ch07/fileupload01.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>File Upload</title>
05 </head>
06 <body>
07     <form name="fileForm" method="post" enctype="multipart/form-data"
08         action="fileupload01_process.jsp">
09         <p>이 름 : <input type="text" name="name">
10         <p>제 목 : <input type="text" name="subject">
11         <p>파 일 : <input type="file" name="filename">
12         <p><input type="submit" value="파일 올리기">
13     </form>
14 </body>
15 </html>
```

## 2. MultipartRequest를 이용한 파일 업로드

JSPBook/WebContent/ch07/fileupload01\_process.jsp

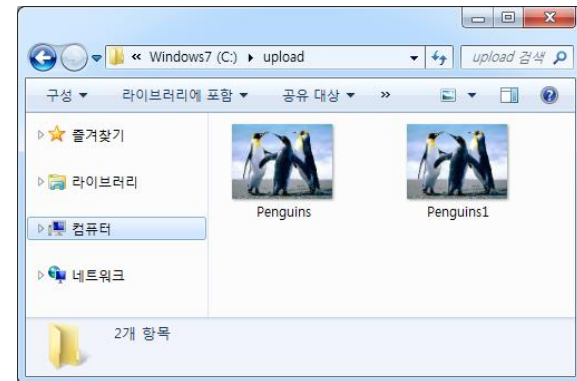
```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="com.oreilly.servlet.*"%>
03 <%@ page import="com.oreilly.servlet.multipart.*"%>
04 <%@ page import="java.util.*"%>
05 <%@ page import="java.io.*"%>
06 <%
07     MultipartRequest multi = new MultipartRequest(request, "C:\\upload",
08         5 * 1024 * 1024, "utf-8", new DefaultFileRenamePolicy());
09     Enumeration params = multi.getParameterNames();
10
11     while (params.hasMoreElements()) {
12         String name = (String) params.nextElement();
13         String value = multi.getParameter(name);
14         out.println(name + " = " + value + "<br>");
15     }
16     out.println("-----<br>");
17
18     Enumeration files = multi.getFileNames();
19
20     while (files.hasMoreElements()) {
21         String name = (String) files.nextElement();
22         String filename = multi.getFilesystemName(name);
23         String original = multi.getOriginalFileName(name);
24         String type = multi.getContentType(name);
25         File file = multi.getFile(name);
26
27         out.println("요청 파라미터 이름 : " + name + "<br>");
28         out.println("실제 파일 이름 : " + original + "<br>");
29         out.println("저장 파일 이름 : " + filename + "<br>");
30         out.println("파일 콘텐츠 유형 : " + type + "<br>");
31
32         if (file != null) {
33             out.println("파일 크기 : " + file.length());
34             out.println("<br>");
35         }
36     }
37 %>
```



name = 홍길순  
subject = 펭귄

---

요청 파라미터 이름 : filename  
실제 파일 이름 : Penguins.jpg  
저장 파일 이름 : Penguins1.jpg  
파일 콘텐츠 타입 : image/jpeg  
파일 크기 : 777835



## 2. MultipartRequest를 이용한 파일 업로드

### 예제 7-2 MultipartRequest 클래스를 이용하여 여러 파일 업로드 및 정보 출력하기

JSPBook/WebContent/ch07/fileupload02.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>File Upload</title>
05 </head>
06 <body>
07     <form name="fileForm" method="post" enctype="multipart/form-data"
08         action="fileupload02_process.jsp">
09         <p>이 름1 : <input type="text" name="name1">
10         제 목1 : <input type="text" name="subject1">
11         파 일1 : <input type="file" name="filename1">
12     <p>이 름2 : <input type="text" name="name2">
13     제 목2 : <input type="text" name="subject2">
14     파 일2 : <input type="file" name="filename2">
15     <p>이 름3 : <input type="text" name="name3">
16     제 목3 : <input type="text" name="subject3">
17     파 일3 : <input type="file" name="filename3">
18     <p> <input type="submit" value="파일 올리기">
19 </form>
20 </body>
21 </html>
```

## 2. MultipartRequest를 이용한 파일 업로드

JSPBook/WebContent/ch07/fileupload02\_process.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="com.oreilly.servlet.*"%>
03 <%@ page import="com.oreilly.servlet.multipart.*"%>
04 <%@ page import="java.util.*"%>
05 <%@ page import="java.io.*"%>
06 <html>
07 <head>
08 <title>File Upload</title>
09 </head>
10 <body>
11     <%
12         MultipartRequest multi = new MultipartRequest(request, "C:\\upload",
13             5 * 1024 * 1024, "utf-8", new DefaultFileRenamePolicy());
14
15         String name1 = multi.getParameter("name1");
16         String subject1 = multi.getParameter("subject1");
17
18         String name2 = multi.getParameter("name2");
19         String subject2 = multi.getParameter("subject2");
20
21         String name3 = multi.getParameter("name3");
22         String subject3 = multi.getParameter("subject3");
23
24         Enumeration files = multi.getFileNames();
25
26         String file3 = (String) files.nextElement();
27         String filename3 = multi.getFilesystemName(file3);
```



## 2. MultipartRequest를 이용한 파일 업로드

```
28     String file2 = (String) files.nextElement();
29     String filename2 = multi.getFilesystemName(file2);
30
31     String file1 = (String) files.nextElement();
32     String filename1 = multi.getFilesystemName(file1);
33     %>
34     <table border="1">
35         <tr>
36             <th width="100">이름</th>
37             <th width="100">제목</th>
38             <th width="100">파일</th>
39         </tr>
40         <%
41             out.print("<tr><td>" + name1 + " </td>");
42             out.print("<td>" + subject1 + " </td>");
43             out.println("<td> " + filename1 + "</td></tr>\n");
44
45             out.print("<tr><td>" + name2 + " </td>");
46             out.print("<td>" + subject2 + " </td>");
47             out.println("<td> " + filename2 + "</td></tr>\n");
48
49             out.print("<tr><td>" + name3 + " </td>");
50             out.print("<td>" + subject3 + " </td>");
51             out.println("<td> " + filename3 + "</td></tr>\n");
52         <%>
53     </table>
54 </body>
55 </html>
```

File Upload

http://localhost:8080/JSPBook/ch07/fileupload02.jsp

이름1: 홍길순    제목1: 튜립    파일1: C:\Tulips.jpg    찾아보기...

이름2: 홍길동    제목2: 코알라    파일2: C:\Koala.jpg    찾아보기...

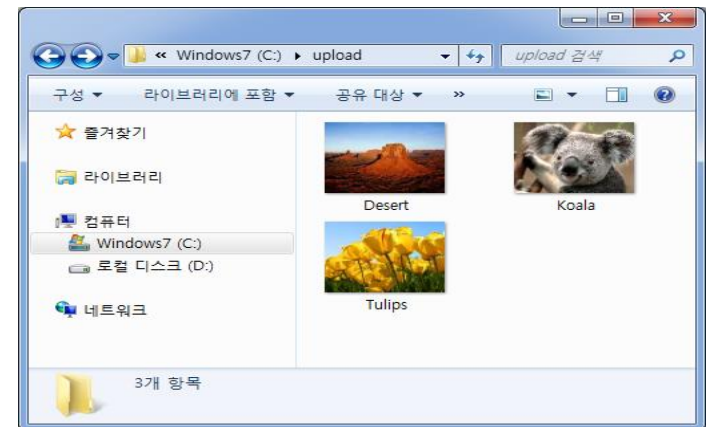
이름3: 수여인    제목3: 사막    파일3: C:\Desert.jpg    찾아보기...

파일 올리기

File Upload

/ch07/fileupload02\_process.jsp

이름	제목	파일
홍길순	튜립	Tulips.jpg
홍길동	코알라	Koala.jpg
수여인	사막	Desert.jpg



# 3. Commons-FileUpload를 이용한 파일 업로드

## ❖ Commons-FileUpload

- 파일 업로드 패키지
- 서버의 메모리상에서 파일 처리가 가능하도록 지원
- 오픈 라이브러리 commons-fileupload.jar, commons-io.jar 파일을 배포 사이트에서 직접 다운로드해서 사용
  - 배포 사이트: <http://commons.apache.org/downloads/>
  - 다운로드 파일: commons-fileupload-1.3.3-bin.zip, commons-io-2.6-bin.zip
- JSP 페이지에 page 디렉티브 태그의 import 속성을 사용하여 패키지 org.apache.commons.fileupload.\*를 설정

### NOTE\_ 오픈 라이브러리 Commons-FileUpload.jar, commons-io.jar 설치 방법

- ① 배포 사이트에서 commons-fileupload-1.3.3-bin.zip와 commons-io-2.6-bin.zip 파일을 다운로드합니다.
- ② 다운로드한 두 파일의 압축을 풉니다.
- ③ 압축을 푼 두 파일의 commons-fileupload-1.3.3-bin, commons-io-2.6-bin 폴더에 있는 commons-fileupload-1.3.2.jar, commons-io-2.6.kar 파일을 찾습니다.
- ④ 웹 애플리케이션의 /WebContent/WEB-INF/lib 폴더에 commons-fileupload-1.3.2.jar, commons-io-2.6.kar 파일을 포함합니다. 또는 서버 톰캣의 /common/lib 폴더와 자바 JDK의 lib 폴더에 commons-fileupload-1.3.2.jar, commons-io-2.6.kar 파일을 포함합니다.

# 3. Commons-FileUpload를 이용한 파일 업로드

```
<%page contentType="text/html; charset=utf-8"%>
<%page import="org.apache.commons.fileupload.*"%>
<%page import="java.io.*"%>
<%page import="java.util.*"%>
```

```
<html>
<head>
<title>File Upload</title>
</head>
<body>
<%
```

```
String fileUploadPath = "C:\\upload";
```

```
DiskFileUpload upload = new DiskFileUpload();
```

```
List items = upload.parseRequest(request);
```

```
Iterator params = items.iterator();
```

```
while (params.hasNext()) {
```

```
    FileItem item = (FileItem) params.next();
    if (item.isFormField()) {
        String title = item.getString("UTF-8");
        out.println("<h3>" + title + "</h3>");
    } else {
        String fileName = item.getName();
        fileName = fileName.substring(fileName.lastIndexOf("\\") + 1);
        File file = new File(fileUploadPath + "/" + fileName);
        item.write(file);
        out.println("파일 이름 : " + fileName + "<br>");
    }
}
```

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

제목 : 파일 업로드

파일 : C:\Penguins.jpg

찾아보기...

submit

파일 업로드

파일 이름 : Penguins.jpg

# 3. Commons-FileUpload를 이용한 파일 업로드

## ■ DiskFileUpload 클래스의 메소드

메소드	유형	설명
setRepositoryPath(String repositoryPath)	void	업로드된 파일을 임시로 저장할 디렉터리를 설정합니다.
setSizeMax(long sizeMax)	void	최대 파일의 크기를 설정합니다.
setSizeThreshold(int sizeThreshold)	void	메모리상에 저장할 최대 크기를 설정합니다.
parseRequest(HttpServletRequest req)	List<FileItem>	multipart/form-data 유형의 요청 파라미터를 가져옵니다.

## ■ FileItem 클래스의 메소드

메소드	유형	설명
isFormField()	boolean	요청 파라미터가 파일이 아니라 일반 데이터인 경우 true를 반환합니다.
getFieldName()	String	요청 파라미터의 이름을 얻어옵니다.
getString()	String	기본 문자 인코딩을 사용하여 요청 파라미터의 값을 얻어옵니다.
getString(String encoding)	String	설정된 문자 인코딩을 사용하여 요청 파라미터의 값을 얻어옵니다.
getName()	String	업로드된 파일(경로 포함)의 이름을 얻어옵니다.
getSize()	long	업로드된 파일의 크기를 얻어옵니다.
get()	byte[]	업로드된 파일을 바이트 배열로 얻어옵니다.
isInMemory()	boolean	업로드된 파일이 메모리에 저장된 상태인 경우 true를 반환하고, 임시 디렉터리에 저장된 경우 false를 반환합니다.
delete()	void	파일과 관련된 자원을 삭제합니다. 메모리상에 저장된 경우 할당된 메모리를 반환하고, 임시 파일로 저장된 경우 파일을 삭제합니다.
write()	void	파일과 관련된 자원을 저장합니다.
getContentType()	String	웹 브라우저가 전송하는 콘텐츠 유형을 반환하고, 정의되어 있지 않은 경우 null을 반환합니다.

### 3. Commons-FileUpload를 이용한 파일 업로드

#### 예제 7-3 Commons-FileUpload를 이용하여 파일 업로드하기

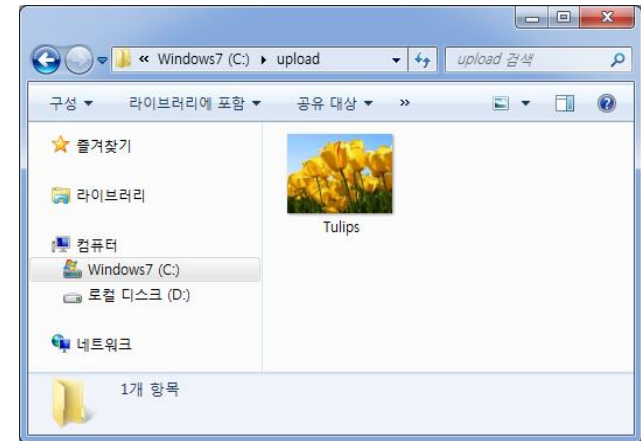
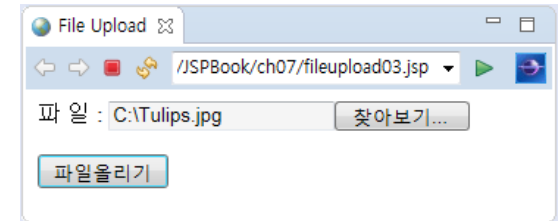
JSPBook/WebContent/ch07/fileupload03.jsp

```
01 <%page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>File Upload</title>
05 </head>
06 <body>
07     <form action="fileupload03_process.jsp" method="post"
08         enctype="multipart/form-data">
09         <p>파 일 : <input type="file" name="filename">
10         <p><input type="submit" value="파일 올리기">
11     </form>
12 </body>
13 </html>
```

# 3. Commons-FileUpload를 이용한 파일 업로드

JSPBook/WebContent/ch07/fileupload03\_process.jsp

```
01 <%page contentType="text/html; charset=utf-8"%>
02 <%page import="org.apache.commons.fileupload.*"%>
03 <%page import="java.util.*"%>
04 <%page import="java.io.*"%>
05 <html>
06 <head>
07 <title>File Upload</title>
08 </head>
09 <body>
10     <%
11         String fileUploadPath = "C:\\upload";
12
13         DiskFileUpload upload = new DiskFileUpload();
14
15         List items = upload.parseRequest(request);
16
17         Iterator params = items.iterator();
18
19         while (params.hasNext()) {
20             FileItem fileItem = (FileItem) params.next();
21             if (!fileItem.isFormField()) {
22                 String fileName = fileItem.getName();
23                 fileName = fileName.substring(fileName.lastIndexOf("\\") + 1);
24                 File file = new File(fileUploadPath + "/" + fileName);
25                 fileItem.write(file);
26             }
27         }
28     %>
29 </body>
30 </html>
```

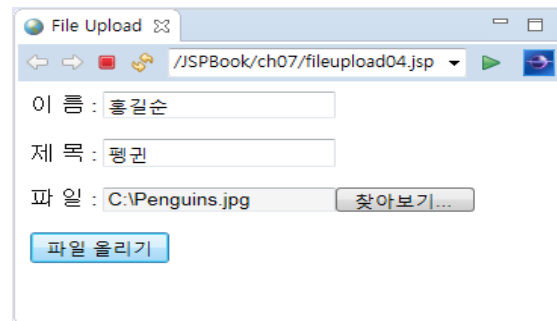


### 3. Commons-FileUpload를 이용한 파일 업로드

#### 예제 7-4 Commons-FileUpload를 이용하여 파일 업로드 및 정보 출력하기

JSPBook/WebContent/ch07/fileupload04.jsp

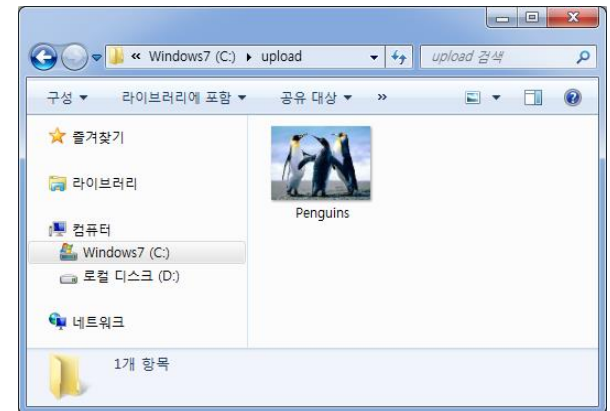
```
01 <% page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <title>File Upload</title>
05 </head>
06 <body>
07     <form name="fileForm" method="post" enctype="multipart/form-data"
        action="fileupload04_process.jsp">
08         <p>이 름 : <input type="text" name="name">
09         <p>제 목 : <input type="text" name="subject">
10         <p>파 일 : <input type="file" name="filename">
11         <p><input type="submit" value="파일 올리기">
12     </form>
13 </body>
14 </html>
```



# 3. Commons-FileUpload를 이용한 파일 업로드

JSPBook/WebContent/ch07/fileupload04\_process.jsp

```
01 <%page contentType="text/html; charset=utf-8"%>
02 <%page import="org.apache.commons.fileupload.*"%>
03 <%page import="java.io.*"%>
04 <%page import="java.util.*"%>
05 <html>
06 <head>
07 <title>File Upload</title>
08 </head>
09 <body>
10     <%
11         String path = "C:\\upload";
12
13         DiskFileUpload upload = new DiskFileUpload();
14
15         upload.setSizeMax(1000000);
16         upload.setSizeThreshold(4096);
17         upload.setRepositoryPath(path);
18
19         List items = upload.parseRequest(request);
20         Iterator params = items.iterator();
21
22         while (params.hasNext()) {
23             FileItem item = (FileItem) params.next();
24
```

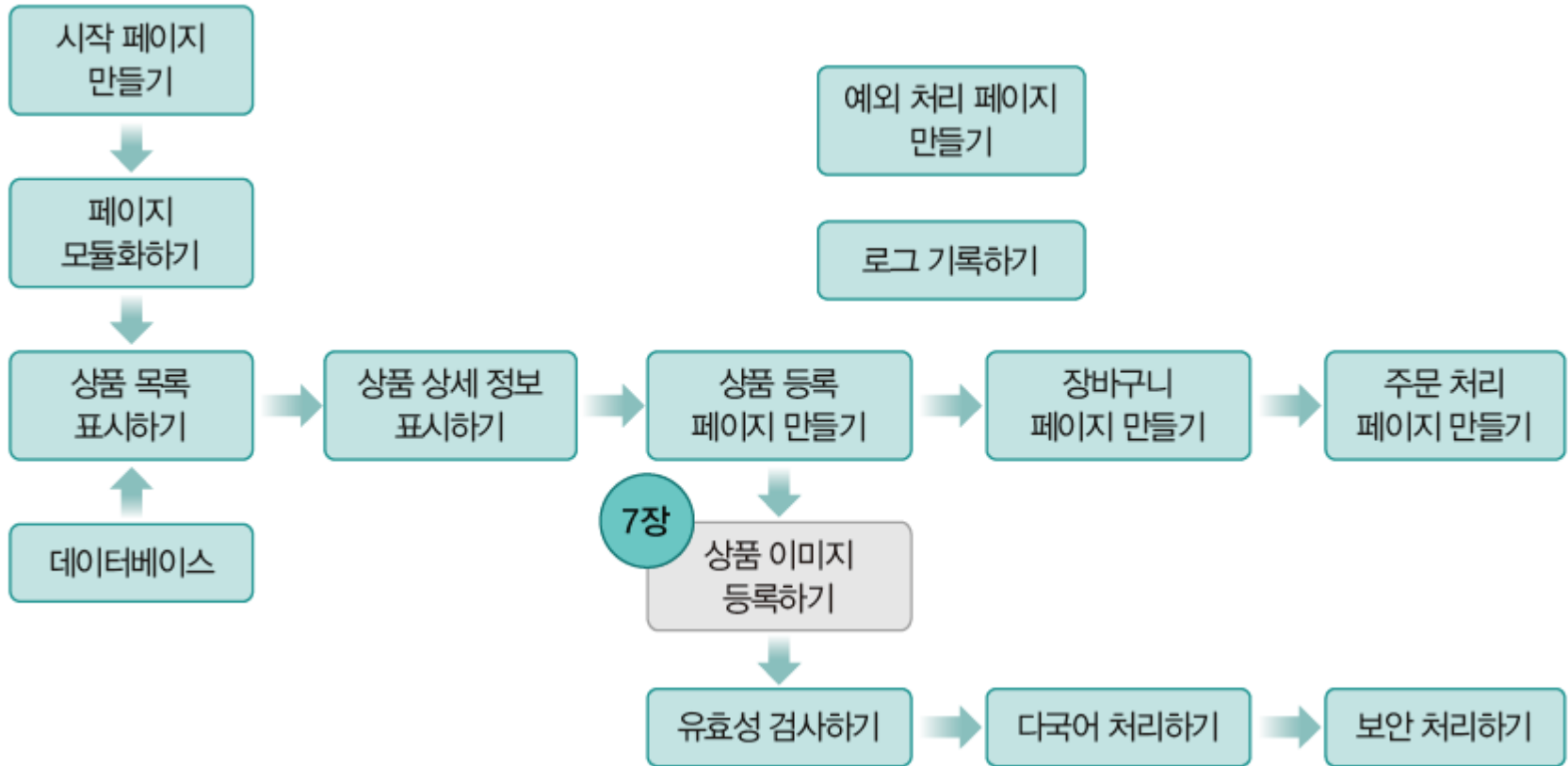




### 3. Commons-FileUpload를 이용한 파일 업로드

```
25         if (item.isFormField()) {
26             String name = item.getFieldName();
27             String value = item.getString("utf-8");
28             out.println(name + "=" + value + "<br>");
29         } else {
30             String fileFieldName = item.getFieldName();
31             String fileName = item.getName();
32             String contentType = item.getContentType();
33
34             fileName = fileName.substring(fileName.lastIndexOf("\\") + 1);
35             long fileSize = item.getSize();
36
37             File file = new File(path + "/" + fileName);
38             item.write(file);
39
40             out.println("—————<br>");
41             out.println("요청 파라미터 이름 : " + fileFieldName + "<br>");
42             out.println("저장 파일 이름 : " + fileName + "<br>");
43             out.println("파일 콘텐츠 유형 : " + contentType + "<br>");
44             out.println("파일 크기 : " + fileSize);
45         }
46     }
47     %>
48 </body>
49 </html>
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기



## 4. [웹 쇼핑몰] 상품 이미지 등록하기

상품 등록

http://localhost:8080/WebMarket/addProduct.jsp

Home

상품 등록

상품 코드

P1237

상품명

Galaxy Note5

가격

900000

상세 정보

212 8\*126 6\*6mm, Super AMOLED display, Octa-Core processor

제조사

삼성

분류

핸드폰

재고 수

100

상태

☒ 신규 제품 ☐ 중고 제품 ☐ 재생 제품

이미지

C:\P1237.jpg

찾아보기...


등록

상품 목록


http://localhost:8080/WebMarket/products.jsp

Home


상품 목록




**iPhone 6s**  
4.7-inch, 1334X750 Retina HD display, 8-megapixel iSight Camera  
800000원  
상세 정보 >>



**LG PC 그램**  
13.3-inch, IPS LED display, 5rd Generation Intel Core processors  
1500000원  
상세 정보 >>



**Galaxy Tab S**  
212.8\*125.6\*6.6mm, Super AMOLED display, Octa-Core processor  
900000원  
상세 정보 >>



**Galaxy Note5**  
212 8\*126 6\*6mm, Super AMOLED display, Octa-Core processor  
900000원  
상세 정보 >>

© WebMarket

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### 예제 7-5 정적 리소스(상품 이미지 및 부트스트랩 CSS) 관리하기

#### ■ 상품 클래스에 멤버 변수 추가하기

WebMarket/src/dto/Product.java

```
01 package dto;
02
03 import java.io.Serializable;
04
05 public class Product implements Serializable{
06     ...(생략)...
07     private String condition;    //신상품 or 중고품 or 재생품
08     private String filename;    //이미지 파일명
09 }
10
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

- 추가된 멤버 변수의 Setter/Getter( ) 메소드 작성하기

WebMarket/src/dto/Product.java

```
01 package dto;
02
03 import java.io.Serializable;
04
05 public class Product implements Serializable{
06     ...(생략)...
07     public String getFilename() {
08         return filename;
09     }
10
11     public void setFilename(String filename) {
12         this.filename = filename;
13     }
14 }
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

- 상품 데이터 접근 클래스 수정하기

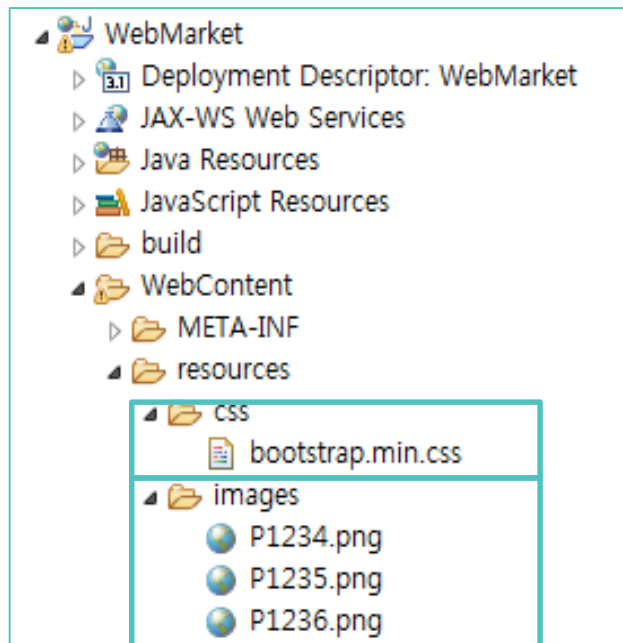
WebMarket/src/dao/ProductRepository.java

```
01 package dao;
02 ...생략...
03 public class ProductRepository {
04     ...생략...
05     public ProductRepository() {
06         ...생략...
07         phone.setCondition("New");
08         phone.setFilename("P1234.png");
09         ...생략...
10         notebook.setCondition("Refurbished");
11         notebook.setFilename("P1235.png");
12         ...
13         tablet.setCondition("Old");
14         tablet.setFilename("P1236.png");
15     ...생략...
16     }
17     ...생략...
18 }
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### ■ 정적 리소스 관리 폴더 만들기

- /WebContent/ 폴더에 resources 폴더를 생성하고, 여기에 이미지
- 파일을 관리하는 images 폴더와 부트스트랩 파일(bootstrap.css)을 관리하는 css 폴더를 생성
- 생성된 image 폴더에 이미지 파일명을 상품 아이디로 하여 등록
- css 폴더에 부트스트랩 CSS 파일(bootstrap.min.css)을 다운로드하여 등록



#### NOTE\_ 부트스트랩 CSS 파일 설치 방법

- 1 배포 사이트(<http://getbootstrap.com/>)에서 bootstrap-4.0.0-dist.zip 파일을 다운로드합니다.
- 2 bootstrap-4.0.0-dist.zip 파일의 압축을 풉니다.
- 3 압축을 푼 파일의 css 폴더에 있는 bootstrap.min.css 파일을 찾습니다.
- 4 웹 애플리케이션의 /WebContent/resources/css/ 폴더에 bootstrap.min.css 파일을 포함합니다.

등록된 부트스트랩 CSS 파일

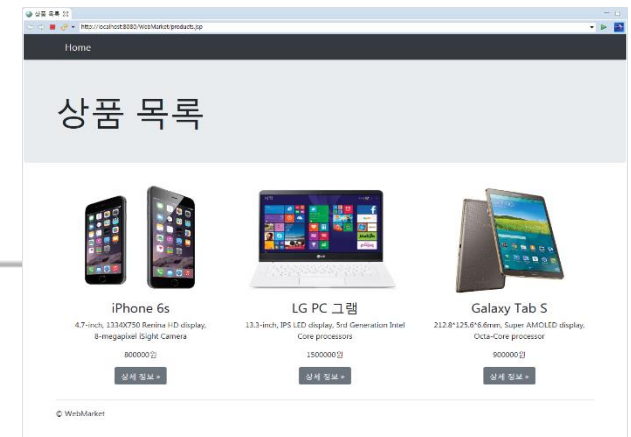
등록된 이미지 파일

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### ■ 상품 목록 페이지 수정하기

WebMarket/WebContent/products.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
04 <title>상품 목록</title>
05 ... (생략) ...
06         <div class="col-md-4">
07             
09             <h3>%=product.getPname()%</h3>
10             <p>%=product.getDescription()%
```



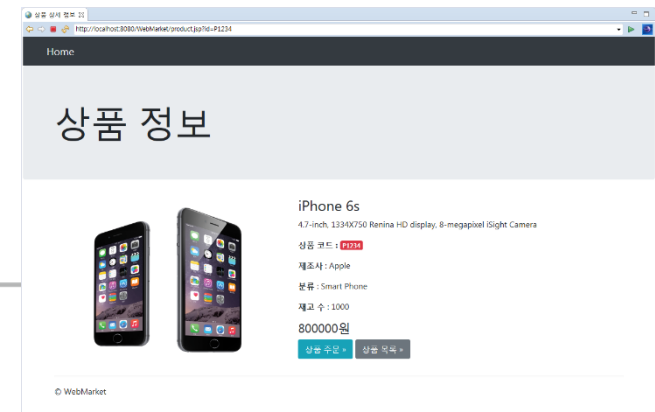


## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### ■ 상품 상세 정보 페이지 수정하기

WebMarket/WebContent/product.jsp

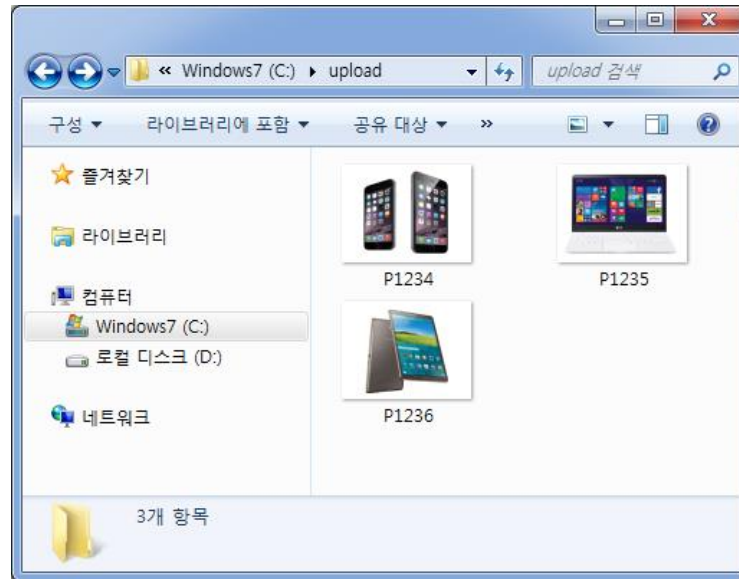
```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03 <head>
04 <link rel="stylesheet" href="./resources/css/bootstrap.min.css" />
05 <title>상품 상세 정보</title>
06 ... (생략) ...
07 <div class="container">
08     <div class="row">
09         <div class="col-md-5">
10             
12         </div>
13         <div class="col-md-6">
14             <h3><%=product.getPname()%></h3>
15             ... (생략) ...
```



## 4. [웹 쇼핑몰] 상품 이미지 등록하기

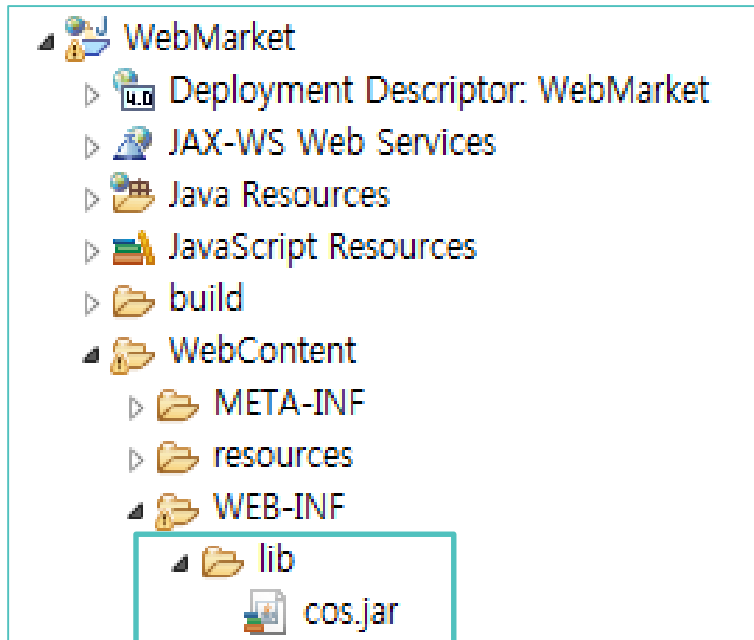
### 예제 7-6 상품 이미지 업로드하기

- 상품 이미지 파일의 저장 위치 만들기
  - C 드라이브에 upload 폴더를 생성하고 이 폴더에 상품 이미지 파일명을 상품 아이디로 하여 등록



## 4. [웹 쇼핑몰] 상품 이미지 등록하기

- 오픈 라이브러리 cos.jar 등록하기



## 4. [웹 쇼핑몰] 상품 이미지 등록하기

- 상품 목록 페이지 수정하기

WebMarket/WebContent/products.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03         <div class="col-md-4">
04             
05                 <h3><%=product.getPname()%></h3>
06                 <p><%=product.getDescription()%>
07
08 ... (생략) ...
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

- 상품 상세 정보 페이지 수정하기

WebMarket/WebContent/product.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 ... (생략) ...
03     <div class="container">
04         <div class="row">
05             <div class="col-md-5">
06                 
07             </div>
08             <div class="col-md-6">
09                 <h3><%=product.getPname()%></h3>
10                 <p><%=product.getDescription()%>
11             ... (생략) ...
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### ■ 상품 정보 등록 페이지 수정하기

WebMarket/WebContent/addProduct.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <html>
03 <head>
04 <link rel="stylesheet" href="../resources/css/bootstrap.min.css" />
05 <title>상품 등록</title>
06 ... (생략) ...
07 <div class="container">
08     <form name="newProduct" action="../processAddProduct.jsp"
09         class="form-horizontal" method="post" enctype="multipart/form-data">
10         ... (생략) ...
11         <div class="form-group row">
12             <label class="col-sm-2">상태</label>
13             <div class="col-sm-5">
14                 <input type="radio" name="condition" value="New ">
15                 신규 제품 <input
16                     type="radio" name="condition" value="Old"> 중고
17                 제품 <input
18                     type="radio" name="condition" value="Refurbished">
19                 재생 제품
20             </div>
21         </div>
22         <div class="form-group row">
23             <label class="col-sm-2">이미지</label>
24             <div class="col-sm-5">
25                 <input type="file" name="productImage" class="form-control">
26             </div>
27         </div>
28     </form>
29 </div>
30 ... (생략) ...
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

### ■ 상품 등록 처리 페이지 수정하기

WebMarket/WebContent/processAddproduct.jsp

```
01 <%@ page contentType="text/html; charset=utf-8"%>
02 <%@ page import="com.oreilly.servlet.*"%>
03 <%@ page import="com.oreilly.servlet.multipart.*"%>
04 <%@ page import="java.util.*"%>
05 <%@ page import="dto.Product"%>
06 <%@ page import="dao.ProductRepository"%>
07 <%
08     request.setCharacterEncoding("UTF-8");
09
10     String filename = "";
11     String realFolder = "C:\\upload"; //웹 애플리케이션상의 절대 경로
12     int maxSize = 5 * 1024 * 1024; //최대 업로드될 파일의 크기 5MB
13     String encType = "utf-8"; //인코딩 유형
14
15     MultipartRequest multi = new MultipartRequest(request, realFolder,
16         maxSize, encType, new DefaultFileRenamePolicy());
17
18     String productId = multi.getParameter("productId");
19     String name = multi.getParameter("name");
20     String unitPrice = multi.getParameter("unitPrice");
21     String description = multi.getParameter("description");
22     String manufacturer = multi.getParameter("manufacturer");
23     String category = multi.getParameter("category");
24     String unitsInStock = multi.getParameter("unitsInStock");
25     String condition = multi.getParameter("condition");
26
27     Integer price;
28     if (unitPrice.isEmpty())
```

## 4. [웹 쇼핑몰] 상품 이미지 등록하기

```
29         price = 0;
30     else
31         price = Integer.valueOf(unitPrice);
32
33     long stock;
34
35     if (unitsInStock.isEmpty())
36         stock = 0;
37     else
38         stock = Long.valueOf(unitsInStock);
39
40     Enumeration files = multi.getFileNames();
41     String fname = (String) files.nextElement();
42     String fileName = multi.getFilesystemName(fname);
43
44     ProductRepository dao = ProductRepository.getInstance();
45
46     Product newProduct = new Product();
47     newProduct.setProductId(productId);
48     newProduct.setPname(name);
49     newProduct.setUnitPrice(price);
50     newProduct.setDescription(description);
51     newProduct.setManufacturer(manufacturer);
52     newProduct.setCategory(category);
53     newProduct.setUnitsInStock(stock);
54     newProduct.setCondition(condition);
55     newProduct.setFilename(fileName);
56
57     dao.addProduct(newProduct);
58
59     response.sendRedirect("products.jsp");
60 %>
```



