

파일 첨부

<파일 업로드 라이브러리 설치>

파일 첨부을 위해서는 라이브러리가 필요!

- ➔ <http://www.servlets.com>에 접속해서 좌측 메뉴의 COS File Upload Library 클릭
- ➔ webpro 밑에 download 파일에 cos-22.05.zip 다운로드후 압축해제
- ➔ cos-22.05/lib 에 있는 cos.jar을 WEB-INF/lib에 복사
- ➔ Java Resources/Libraries에 Web App Libraries가 새로 생긴 것을 확인할 수 있음

이제는 파일 첨부(=업로드) 기능을 구현이 가능해졌다!

<파일 업로드 프로그래밍>

cf) 파일 업로드를 하기 전 WebContent 안에 파일(=이미지)이 등록될 폴더를 따로 만들자!

cf) 파일은 서버에 저장되고 DB에는 파일이름이 저장되기 때문에 DTO에 들어갈 변수는 String!

cf) 한 개 파일 업로드와 여러 개 파일 업로드 방법이 조금 다르니 교안 참조할 것!

- ex1.bookRegisterForm.jsp

```
<form action="bookRegisterPro.jsp" method="post" enctype="multipart/form-data">
```

파일을 넘길 때는 method 속성의 속성값을 post로 설정

enctype 속성의 속성값을 multipart/form-data로 설정

-> 해당 방식은 <form> 요소가 파일이나 이미지를 서버로 전송할 때 주로 사용.

```
<td><input type="file" name="book_image1"></td>
```

cf) file 타입은 자바빈이 인식하지 못하기 때문에 만약 name을 DTO의 속성명과 지정했다면 바꿔주는 것이 좋다.

- ➔ file 및 Date 타입은 따로 getParameter()로 받아서 형변환을 거친 뒤에 세팅해준다!

- ex2_bookRegisterPro.jsp

```
String path = request.getRealPath("파일이 등록될 폴더 이름");
```

- ➔ 저장할 path를 설정

```
int size = 5*1024*1024;
```

- ➔ 최대 업로드 용량 설정(5MB)

```
String[] file ={"", ""};
```

- ➔ `getFileSystemName()`로 업로드 된 파일들의 이름을 받아서 넣을 file 배열
그러나 파일 첨부를 한 개만 할 경우는 배열이 아닌 빈스트링("")으로 받는다

```
imageUp = new MultipartRequest(request,path,  
                                size,"utf-8", new DefaultFileRenamePolicy());  
Enumeration<String> files = imageUp.getFileNames();  
int idx = 0;  
while(files.hasMoreElements()){  
    String paramname = files.nextElement(); => 파라미터 이름  
    file[idx] = imageUp.getFileSystemName(paramname); => 서버에 저장된 파일이름  
    idx++;  
}
```

- ➔ `MultipartRequest` 가 매개변수로 받는 `new DefaultFileRenamePolicy()` 는
중복되는 파일 이름이 있을 때 덮어씌워지는 것을 방지

- ➔ `Enumeration<String>` 은 객체들을 집합체 형태로 관리하게 해주는 인터페이스!
논리적인 커서가 존재하여 커서를 이동하면서 데이터를 꺼내옴.

- `hasMoreElements()` 은 `Enumeration` 과 함께 쓰이는 메소드로 Boolean 타입을 반환
현재 커서 이후에 요소들이 있는지 여부를 체크한다. 있으면 true, 없으면 false
- `nextElement()` 도 `Enumeration` 과 함께 쓰이는 메소드로 `Enumeration<String>`
이 String 타입의 객체를 관리함으로 해당 메소드도 String 타입을 반환
커서를 다음 요소로 이동시키고, 가리키고 있는 요소 객체를 꺼내 반환!

----- 여기까지 <파일 업로드> -----

```
for(String f : file<=배열이름>){ => 파일 첨부 한 개만 할 경우 for문 필요 X  
    InputStream is = null; => 파일을 읽어 들이는 InputStream  
    OutputStream os = null; => 파일을 쓰는 OutputStream  
    try{  
        File serverFile = new File(path+"/"+f);  
        if(serverFile.exists()){ => 첨부한 파일이 없으면 파일 카피 안함  
            is = new FileInputStream(serverFile); => 원본 파일  
            os = new FileOutputStream("파일이 등록될 폴더 경로 /"+f);  
            ⇨ 복사될 파일  
            byte[] bs = new byte[(int)serverFile.length()];  
            while(true){  
                int nReadByte = is.read(bs);  
                if(nReadByte==-1) break;  
                os.write(bs, 0, nReadByte);  
            }  
        }  
        int nReadByte = is.read(bs);
```

- ➔ `read()` 메소드는 `InputStream`로부터 한 개의 문자(2byte)를 읽고 int 타입을 리턴

```
if(nReadByte==-1) break;
```

➔ InputStream으로부터 문자를 다 읽었다면 read() 메소드가 -1을 리턴

Break를 통해서 while문을 빠져나옴

```
os.write(bs, 0, nReadByte);
```

➔ write()는 OutputStream 객체의 메소드로 문자를 쓴다

bs는 Byte배열 / '0' 은 처음부터 / 'rReadByte' 는 읽어들이는 바이트 수

```
finally{  
    try{  
        if(os!=null) os.close();  
        if(is!=null) is.close();  
    }catch(Exception e){}  
}
```

➔ finally절 안에 OutputStream 객체 / InputStream 순으로 닫으면 끝

----- 여기까지 <파일 카피> -----