

# [upload] 파일업로드 처리시 자체 프레임웍으로 처리 방안 가이드

파일 업로드(싱글, 멀티) 처리시 웹스퀘어 자체 처리가 아닌 프레임웍단으로 넘어가서 처리 필요한 경우 가이드

첨부하여 드리는 [uploadTest5\\_1.xml](#), [uploadTest.jsp](#), [TestController3.java](#) 참조부탁드립니다.

웹스퀘어에서 업로드 또는 멀티업로드 컴퍼넌트의 PROPERTY 중에 액션 항목이 있으며 이를 이용해 처리하는 URL을 지정할 수 있습니다. 즉, upload component 에서 액션 값을 지정하고 업로드 하면 action으로 지정된 JSP/서블릿 등 에 대해 멀티파트 리퀘스트 서브미션이 일어나게 되고 해당 JSP/서블릿 등 에서 이를 처리할 수 있습니다.

서브미션을 처리하고 나서 응답(javascript)을 주게 되면 화면에서는 javascript callback 이 수행 됩니다.

첨부한 파일은 업로드 컴포넌트 웹스퀘어 샘플 입니다. (\* 첨부 jsp 파일은 JSP로 응답을 처리하는 예제 입니다. 멀티파트 리퀘스트 처리는 생략되어 있습니다.)

아래는 업로드 컴포넌트에서 지정된 액션에 대해 스프링 컨트롤러에서 응답을 처리하는 샘플 입니다.

1.화면에서의 action 설정예

```
<w2:multiupload id="multiupload1" maxcount="" maxsize=""
ev:ondone="multiupload1_ondone"
style="position: relative; width: 500px; height: 150px;"
action="/upload2.do" uploadButton="false" filter="" wmode="false"
mode="flash">
</w2:multiupload>
```

2.화면에서 setParam 설정예

```
var cnt = multiupload1.getFileCount();

for(i=0; i<cnt; i++){
    console.log("cnt:" + i);
    multiupload1.setParam(i, "param" + i, "param" + i + "_value");
}

multiupload1.startUpload();
```

3. 컨트롤러에서의 처리 예

```
@RequestMapping("/upload2.do")
@ResponseBody
public void upload2(HttpServletRequest request, HttpServletResponse res) throws Exception{
    System.out.println("upload2.do start=====");

    SimpleDateFormat mSimpleDateFormat = new SimpleDateFormat("yyyyMMddHHmmssSSS", Locale.KOREA);
    Date currentTime = new Date();

    String uploadPath = "/Users/inswave/upload/";
    String saveFileName = null;
    String uploadFileName = null;
    long uploadFileSize = 0;
```

```

String OriginalfileName = null;

MultipartHttpServletRequest multiPartRequest = (MultipartHttpServletRequest) request;

//param 확인
Enumeration<String> e = multiPartRequest.getParameterNames();

String name = null;
while (e.hasMoreElements()){
    name = e.nextElement().toString();

    System.out.println(name + ":" + request.getParameter(name));
}

Iterator<String> iter = multiPartRequest.getFileNames();

int i = 1;
while(iter.hasNext()) {

    String sNewFileName = mSimpleDateFormat.format ( currentTime ) + String.valueOf(i);

    uploadFileName = iter.next();
    MultipartFile mfile = (MultipartFile) multiPartRequest.getFile(uploadFileName);

    uploadFileSize = mfile.getSize();
    OriginalfileName = mfile.getOriginalFilename();
    OriginalfileName = new String(OriginalfileName.getBytes("8859_1"),"utf-8");

    int index=OriginalfileName.lastIndexOf(".");
    String fileExtension="." + OriginalfileName.substring(index + 1);
    String fileName = OriginalfileName.substring(0,index);

    System.out.println("fileExtension="+fileExtension);
    System.out.println("fileName="+fileName);

    saveFileName = uploadPath + fileName + "_" + sNewFileName + fileExtension;
    i++;
    System.out.println("saveFileName="+saveFileName);

    mfile.transferTo(new File(saveFileName));
}

StringBuilder stb1 = new StringBuilder();

stb1.append("<script>window.onload = dolnit;function dolnit() {");
stb1.append("parent.uploadCallback(₩" "<ret>");
stb1.append("<key>"+uploadPath+"</key>");
stb1.append("<storedFileList>"+saveFileName+"</storedFileList>");
stb1.append("<localfileList>"+OriginalfileName+"</localfileList>");
stb1.append("<fileSizeList>"+uploadFileSize+"</fileSizeList>");
stb1.append("<maxUploadSize></maxUploadSize>");
stb1.append("<deniedList></deniedList>");
stb1.append("<deniedCodeList></deniedCodeList>");
stb1.append("</ret>₩");}</script>");

HttpServletResponseWrapper wrapper = new HttpServletResponseWrapper(res);
wrapper.setContentType("text/html;charset=UTF-8");
wrapper.setHeader("Content-length", "" + stb1.toString().getBytes().length);

```



```

    res.getWriter().print(stb1.toString());
}
}

```

#### 4. web.xml 에 설정 추가

<servlet-mapping>

<servlet-name>sample</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

#### 5. TestController3.java 에서 upload2 의 uploadPath 로컬환경에 맞게 수정

#### 6. 테스트 페이지

업로드 성공시 textarea에 업로드 정보가 출력됩니다.

upload2.do controller 실행

uploadStart

#	파일	크기	상태
1	after1.png	33.8 KB	
2	before1.png	33.6 KB	

추가

삭제

최소

```

0번째 deniedFile===
0번째 deniedFile===
0번째 file===/Users/inswave/upload/after1_201711021047450831.png
0번째 key===/Users/inswave/upload/
0번째 localFile===after1.png
0번째 maxUploadSize===
0번째 size===33.8 KB
1번째 deniedFile===
1번째 deniedFile===
1번째 file===/Users/inswave/upload/before1_201711021047451311.png
1번째 key===/Users/inswave/upload/
1번째 localFile===before1.png
1번째 maxUploadSize===
1번째 size===33.6 KB

```

uploadTest.jsp 로 실행한경우

#	파일	크기	상태

추가

삭제

취소

전송