XPlatform 파일 업/다운로드

샘플소스









**\*샘플소스\***

**[HttpObject]**

var downFolder; //사용자가 지정한 폴더 저장

//upload

function btn\_up\_onclick(obj:Button, e:ClickEventInfo)

{

ProgressBar00.pos = 0;

var vFile\_up = FileDialog00.open("file upload",FileDialog.LOAD,"%MYDOCUMENT%");

if (vFile\_up)

{

var rtn = HttpObject00.upload(edt\_url.text +

"fileUpload.jsp?PATH=up\_11",FileDialog.LOAD,vFile\_up,true);

trace("파일 업로드:" + rtn );

}

}

//download

function btn\_down\_onclick(obj:Button, e:ClickEventInfo)

{

ProgressBar00.pos = 0;

downFolder = FileDialog00.open("file download",FileDialog.SELFOLDER); //사용자 폴더 선택

var vFile\_down = new VirtualFile();

vFile\_down.open(edt\_download.text, VirtualFile.openWrite ); //내문서로 다운로드 받음.

if (vFile\_down)

{

var strFileName = vFile\_down.filename;

var rtn = HttpObject00.download("http://localhost:7657/Test\_1/XP\_SERVICE/fileDownload.jsp?file=" + strFileName,vFile\_down,true);

trace(edt\_url.text + " -- 파일 다운로드:" + rtn);

}

}

function HttpObject00\_onstatus(obj:HttpObject, e:HttpObjStatusEventInfo)

{

//trace("HttpObject00\_onstatus::" + e.status);

ProgressBar00.pos = e.status;

}

function HttpObject00\_onload(obj:HttpObject, e:HttpObjLoadEventInfo)

{

alert("onload >> " + e.errorcode + "/"+ e.errormsg);

if(e.errorcode == 200 || e.errorcode == 0) {

ProgressBar00.pos = 100;

}

if( e.errorcode == 200 ) //업로드 완료

{

alert("upload success ");

edt\_download.value = e.errormsg;

}

if( e.errormsg == "Success" ) //다운로드 완료

{

alert("download success");

//ExtCommon의 readFile메소드로 내문서의 temp파일을 읽은 후 writeFile메소드로 지정한 경로에 다시 파일을 생성해줌.

var DownObj;

var objExt = new ExtCommon;

DownObj = objExt.readFile(system.convertRealPath("%DOCUMENT%")+edt\_download.value,"OBJ");

trace(downFolder+edt\_download.value);

objExt.writeFile(downFolder+"\\"+edt\_download.value, DownObj);

VirtualFile.delete(system.convertRealPath("%DOCUMENT%")+edt\_download.value); //임시파일을 지움.

system.execShell(downFolder); //다운로드 받은 폴더를 열어줌.

}

//trace("errorcode = " + e.errorcode + " \ " + "errormsg = " + e.errormsg );

}

**[FileUpload]**

//업로드 버튼을 클릭시 다운로드

function Button00\_onclick(obj:Button, e:ClickEventInfo)

{

var strUrl = "http://localhost:7657/Test\_1/XP\_SERVICE/fileUpload.jsp?PATH=up\_11";

var bSucc = FileUpload00.upload(strUrl);

//alert(bSucc);

}

//다운로드 버튼을 클릭시 다운로드

function Button01\_onclick(obj:Button, e:ClickEventInfo)

{

var strFileName = ds\_uploadresult.getColumn(0,"fileName");

FileDownload00.downloadfilename = strFileName;

trace("strFileName:"+strFileName);

FileDownload00.downloadurl = "http://172.10.11.70:7657/Test\_1/XP\_SERVICE/fileDownload.jsp?file="+strFileName;

var bSucc = FileDownload00.download();

trace("bSucc >> "+bSucc);

}

function FileUpload00\_onload(obj:FileUpload, e:FileUploadLoadEventInfo)

{

if( e.datasets == null )

{

alert("업로드 실패");

} else {

alert("업로드 성공");

trace(e.datasets[0].saveXML());

//업로드 성공 후 받은 데이터셋에서 해당 파일의 파일명을 데이터셋에 저장.

ds\_uploadresult.setColumn(0,"fileName",e.datasets[0].getColumn(0,"fileName"));

Ds\_Del.setColumn(0,"FileNm",e.datasets[0].getColumn(0,"fileName"));

//FileDown컴포넌트를 클릭시 다운로드 되도록 셋팅.

FileDownload00.text = e.datasets[0].getColumn(0,"fileName");

FileDownload00.downloadfilename = e.datasets[0].getColumn(0,"fileName");

FileDownload00.downloadurl = "http://localhost:7657/Test\_1/XP\_SERVICE/fileDownload.jsp?file="+e.datasets[0].getColumn(0,"fileName");

ds\_uploadresult.copyData(e.datasets[0]);

}

}

function FileDownload00\_onload(obj:FileDownload, e:FileDownloadLoadEventInfo)

{

alert("다운로드 성공");

}

function Button02\_onclick(obj:Button, e:ClickEventInfo)

{

transaction("Del" , "http://localhost:7657/Test\_1/[XP]HttpMultiPart\_delete.jsp?PATH=up\_11", "input=Ds\_Del", "", "", "fn\_callback");

}

function fn\_callback( svcID, nCD , sMSG)

{

if (nCD < 0 )

{

alert("실패");

return;

}

if (svcID == "Del")

{

alert(sMSG);

}

}

**[JSP]**

**FileUpload.jsp**

<%@ page contentType=*"text/html;charset=euc-kr"* %>

<%@ page language=*"java"*%>

<%@ page import=*"java.io.File"*%>

<%@ page import=*"java.io.IOException"*%>

<%@ page import=*"java.util.ArrayList"*%>

<%@ page import=*"java.util.Enumeration"*%>

<%@ page import=*"java.util.Iterator"*%>

<%@ page import=*"java.util.List"*%>

<%@ page import=*"javax.servlet.ServletException"*%>

<%@ page import=*"javax.servlet.http.HttpServletRequest"*%>

<%@ page import=*"javax.servlet.http.HttpServletResponse"*%>

<%@ page import=*"com.oreilly.servlet.MultipartRequest"*%>

<%@ page import=*"com.oreilly.servlet.multipart.DefaultFileRenamePolicy"*%>

<%@ page import=*"com.tobesoft.xplatform.data.ColumnHeader"*%>

<%@ page import=*"com.tobesoft.xplatform.data.DataSet"*%>

<%@ page import=*"com.tobesoft.xplatform.data.DataTypes"*%>

<%@ page import=*"com.tobesoft.xplatform.data.PlatformData"*%>

<%@ page import=*"com.tobesoft.xplatform.data.VariableList"*%>

<%@ page import=*"com.tobesoft.xplatform.tx.HttpPlatformResponse"*%>

<%@ page import=*"com.tobesoft.xplatform.tx.PlatformException"*%>

<%

String chkType = request.getHeader("Content-Type");

//System.out.println(chkType);

**if**( chkType == **null** )

**return**;

request.setCharacterEncoding("utf-8");

String contextRealPath = request.getSession().getServletContext().getRealPath("/");

String PATH = request.getParameter("PATH");

String savePath = contextRealPath + PATH;

System.out.println(savePath);

**int** maxSize = 500 \* 1024 \* 1024; // 최대 업로드 파일 크기 500MB(메가)로 제한

PlatformData resData = **new** PlatformData();

VariableList resVarList = resData.getVariableList();

String sMsg = " A ";

**try** {

MultipartRequest multi = **new** MultipartRequest(request, savePath, maxSize, "utf-8", **new** DefaultFileRenamePolicy());

Enumeration files = multi.getFileNames(); // 파일명 모두 얻기

sMsg += "B ";

DataSet ds = **new** DataSet("Dataset00");

ds.addColumn(**new** ColumnHeader("fileName", DataTypes.STRING));

ds.addColumn(**new** ColumnHeader("fileSize", DataTypes.LONG));

ds.addColumn(**new** ColumnHeader("fileType", DataTypes.STRING));

sMsg += "C ";

String fileName="";

**while** (files.hasMoreElements()) {

sMsg += "D ";

String name = (String)files.nextElement();

fileName += multi.getFilesystemName(name);

String type = multi.getContentType(name);

File f = multi.getFile(name);

System.out.println("fileName:"+fileName);

**int** row = ds.newRow();

ds.set(row, "fileName", fileName);

ds.set(row, "fileType", type);

**if** (f != **null**)

{

ds.set(row, "fileSize", f.length());

}

sMsg += row +" ";

}

resData.addDataSet(ds);

resVarList.add("ErrorCode", 200);

//resVarList.add("ErrorMsg", savePath+"/"+fileName);

resVarList.add("ErrorMsg", fileName);

} **catch** (Exception e) {

resVarList.add("ErrorCode", 500);

resVarList.add("ErrorMsg", sMsg + " " + e);

}

HttpPlatformResponse res = **new** HttpPlatformResponse(response);

res.setData(resData);

res.sendData();

%>

**FileDownload.jsp**

<%@page import=*"java.io.BufferedOutputStream"*%>

<%@ page contentType=*"text/html;charset=euc-kr"* %>

<%@ page language=*"java"*%>

<%@ page import=*"java.io.BufferedInputStream"*%>

<%@ page import=*"java.io.FileInputStream"*%>

<%@ page import=*"java.io.File"*%>

<%@ page import=*"java.io.IOException"*%>

<%@ page import=*"javax.servlet.ServletException"*%>

<%@ page import=*"javax.servlet.ServletOutputStream"*%>

<%@ page import=*"javax.servlet.http.HttpServletRequest"*%>

<%@ page import=*"javax.servlet.http.HttpServletResponse"*%>

<%@ page import=*"com.oreilly.servlet.MultipartRequest"*%>

<%@ page import=*"com.oreilly.servlet.multipart.DefaultFileRenamePolicy"*%>

<%

String contextRealPath = request.getSession().getServletContext().getRealPath("/");

String savePath = contextRealPath + "up\_11";

String name = request.getParameter("file");

String filename = **new** String(name.getBytes("iso8859-1"), "UTF-8");

**byte**[] buffer = **new** **byte**[1024];

BufferedOutputStream out\_stream = **new** BufferedOutputStream(response.getOutputStream());

BufferedInputStream in\_stream = **null**;

File fis = **new** File(savePath + "/" + filename);

**if**(fis.exists()){

**try**{

response.setContentType("utf-8");

response.setContentType("application/octet;charset=utf-8");

response.setHeader("Accept-Ranges", "bytes");

response.setHeader("Content-Disposition", "attachment;filename=" + java.net.URLEncoder.encode(filename,"UTF8"));

out.clear();

out = pageContext.pushBody();

in\_stream = **new** BufferedInputStream(**new** FileInputStream(fis));

**int** n = 0;

**while** ((n = in\_stream.read(buffer, 0, 1024)) != -1) {

out\_stream.write(buffer, 0, n);

}// while

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**if** (in\_stream != **null**) {

**try** {

in\_stream.close();

} **catch** (Exception e) {}

}

**if** (out\_stream != **null**) {

**try** {

out\_stream.close();

} **catch** (Exception e) {}

}

}

}**else**{

response.sendRedirect("unknownfile");

}

%>

# HttpOjbect로 다른 경로의 파일 멀티업로드 가능여부

같은 폴더내의 파일들은 기본적으로 멀티 업로드가 가능합니다.

각각 다른 경로의 파일들을 업로드 하는 경우의 처리 방안입니다.

## 상세설명

upload메소드의 3번째 메소드에 VirtualFile이 담긴 배열도 지원을 하도록 기능이 추가 되었습니다.

### 소스 코드

* + 사용예)

|  |
| --- |
| var vFileArr = new Array(); //virtualfile을 담기위한 배열 선언  vFileArr[0] = vFile\_up; //virtualfile 을 배열에 담음  vFileArr[1] = vFile\_up1; //virtualfile 을 배열에 담음  var rtn = HttpObject00.upload(edt\_url.text+"Upload.jsp?PATH=up\_11",FileDialog.LOAD,vFileArr,true);  trace("파일 업로드:" + rtn ); |

## 질문유형

* 각각 다른 경로의 파일을 멀티로 업로드 할수 있나요?

## 기타

* 접수번호 :
* 조회건수 :

# 업로드시 서비스가 2번 호출되는 경우 처리

## 간략설명

2011년 6월 패치 이후부터 통신방식 변경으로 인하여(wininet) 파일 업로드시 호출이 2번 되게 됩니다.

(권한획득을 위한 호출을 한 후 파일 업로드가 되어지게 됩니다.)

## 상세설명

서버에서 에러로그가 남게 되는데 이에 대한 처리 방법 입니다.

java.io.IOException: Posted content type isn't multipart/form-data

해당 에러는 권한 획득을 위해 첫번째 호출시 Content-type이 설정되지 않은채 가기 때문에

발생되는 에러 입니다.

### 소스 코드

* + 사용예) 업로드 jsp에 아래와 같이 추가하여 주시면 됩니다.

|  |
| --- |
| String chkType = request.getHeader("Content-Type");  //System.out.println(chkType);  //Content-Type이 null 일때 리턴  if( chkType == null )  return; |

## 질문유형

* 파일 업로드시 서버에서 에러가 발생됩니다.

## 기타

* 접수번호 :
* 조회건수 :

# 파일의 URL정보만으로 다운로드 방법입니다.

## 간략설명

파일의 다운로드 URL 정보만 알고 있는 경우 다운로드 방법 입니다.

### 소스 코드

* + 사용예) 업로드 jsp에 아래와 같이 추가하여 주시면 됩니다.

|  |
| --- |
| //내문서로 다운로드 받음. 파일명 지정함.  VirtualFile00.open("Down\_h1\_naver.gif", VirtualFile.openWrite );  HttpObject00.download("http://XXX/h1\_naver.gif",VirtualFile00,true); |

## 질문유형

* 서버에 올라가 있는 파일의 URL 정보로 다운로드 하는 방법이 있나요?

## 기타

* 접수번호 :
* 조회건수 :

# 파일 다운로드시 Confirm창이 뜨는 경우 처리

## 간략설명

SandBox 정책에 의하여 내문서 폴더가 아닌 경로에 접근시에는 사용자에게 접근여부를

확인하는 Confirm창이 뜨게 됩니다.

이를 뜨지 않게 할 수는 없습니다.

이 경우 내문서로 우선 다운로드 하여 지정한 폴더로 이동시켜주는 로직을 추가 하여야 합니다.

## 상세설명

ExtCommon의 readFile메소드로 내문서의 temp파일을 읽은 후 writeFile메소드로 지정한 경로에 다시 파일을 생성해주도록 해야합니다.

## 소스코드

* + 사용예)

|  |
| --- |
| function HttpObject00\_onload(obj:HttpObject, e:HttpObjLoadEventInfo)  {  if( e.errorcode == 200 ) //업로드 완료  {  alert("upload success ");  edt\_download.value = e.errormsg;  }    if( e.errormsg == "Success" ) //다운로드 완료  {  //ExtCommon의 readFile메소드로 내문서의 temp파일을 읽은 후 writeFile메소드로 지정한 경로에 다시 파일을 생성해줌.  var DownObj;  var objExt = new ExtCommon;  DownObj = objExt.readFile(system.convertRealPath("%DOCUMENT%")+edt\_download.value,"OBJ");  objExt.writeFile(downFolder+"\\"+edt\_download.value, DownObj);  VirtualFile.delete(system.convertRealPath("%DOCUMENT%")+edt\_download.value); //임시파일 지움  }  } |

## 질문유형

* 파일 다운로드시 Confirm창을 뜨지 않게 할수 있나여?

## 기타

* 접수번호 :
* 조회건수 :

# 파일 다운로드가 되지 않습니다.

## 간략설명

파일 접근이 가능한 경로는 내문서 폴더가 기본입니다.

내문서 이외에 접근시에는 ADL의 filesecurelevel속성값을 all 로 설정하여야 합니다.

다운로드를 받는 서버가 실행중인 서버의 주소가 IP와 도메인으로 각각 타입이 다를 경우 설정값이 private이면 접근이 허용되지 않습니다.

이 경우에도 all로 설정을 하여야 합니다.

## 질문유형

* 파일 다운로드가 되지 않습니다.

## 기타

* 접수번호 :
* 조회건수 :

# 파일 업로드 후 리턴값을 받는 방법

## 간략설명

파일 업로드 작업 후 서버에서 보내주는 정보를 받을때의 처리 방법 입니다.

## 상세설명

HttpOject 를 이용하여 파일 업로드 시에는 데이터셋을 리턴값으로 받을 수 없습니다.

이 경우 ErrorMsg를 통하여 값을 받아야 합니다.

보내는 값이 여러 개 일 경우는 구분자를 추가하여 하나의 문자열로 붙여서 넘기고

화면단에서는 구분자를 기준으로 잘라서 사용합니다.

FileUpload 를 이용하여 파일 업로드 시에는 데이터셋을 리턴값으로 받을 수 있습니다.

## 상세설명

|  |
| --- |
| [서버]  **while** (files.hasMoreElements()) {  sMsg += "D ";  String name = (String)files.nextElement();  **fileName += multi.getFilesystemName(name);**  String type = multi.getContentType(name);  File f = multi.getFile(name);  System.out.println("fileName:"+fileName);  **int** row = ds.newRow();  ds.set(row, "fileName", fileName);  ds.set(row, "fileType", type);    **if** (f != **null**)  {  ds.set(row, "fileSize", f.length());  }  sMsg += row +" ";  }    resData.addDataSet(ds);  resVarList.add("ErrorCode", 200);  **resVarList.add("ErrorMsg", fileName);**  [화면]  function HttpObject00\_onload(obj:HttpObject, e:HttpObjLoadEventInfo)  {  Trace(e.errormsg); //errormsg로 받습니다.  }  -FileUpload 컴포넌트에서 데이터셋을 리턴값으로 받는 경우 처리  function FileUpload00\_onload(obj:FileUpload, e:FileUploadLoadEventInfo)  {  if( e.datasets == null )  {  alert("업로드 실패");  } else {  alert("업로드 성공");  trace(**e.datasets[0]**.saveXML());  //업로드 성공 후 받은 데이터셋에서 해당 파일의 파일명을 데이터셋에 저장.  ds\_uploadresult.setColumn(0,"fileName",e.datasets[0].getColumn(0,"fileName"));  }  } |

## 질문유형

* 파일 업로드 후 리턴값을 받을 수 있나요?

## 기타

* 접수번호 :
* 조회건수 :

# 업로드시 가능 용량은?

파일 업로드시 용량제한은 jsp에서 지정하시는 부분입니다.

해당 서비스단 페이지에서 설정 하실 수 있습니다.

멀티 업로드시 100MB 이상의 대용량 파일은 되지 않습니다.

## 질문유형

* 파일 업로드 가능 용량은?

## 기타

* 접수번호 :
* 조회건수 :

# 로컬파일의 경로가 fakepath로 나오는 현상

## 간략설명

HTML5버전에서 로컬 파일의 경로를 표현시 fakepath 로 경로명이 나오게 되는데

이는 IE의 보안상 제약사항으로 정상 동작 입니다.

## 상세설명

* 이는 보안상의 문제로 IE의 옵션값을 조절하면 해결이 가능합니다.

해결방법은

1.IE 도구

2.인터넷 옵션 -> 보안 -> 사용자 지정 수준

3.파일을 서버에 업로드할 때 로컬 디렉터리 경로 포함 을 [사용]으로 체크하세요.

파일 업로드시 내부적으로 파일 경로가 참조되므로 파일경로가 fakepath 로 나와도 파일 업로드는 정상적으로 됩니다.

## 질문유형

* 파일의 경로가 fakepath로 나옵니다.

## 기타

* 접수번호 :
* 조회건수 :

# 업로드 실패시에도 TRUE가 리턴되는 현상

## 간략설명

Upload 메소드의 리턴값이 파일 업로드가 실패하였을 경우에도 true가 나옵니다.

## 상세설명

해당 리턴값은 upload메소드의 실행성공 여부입니다.

실제 파일 업로드에 대한 성공 여부가 아닙니다.

파일의 성공여부를 체크하고자 하시면 onload 이벤트에서 아래와 같이 체크 하시기 바랍니다.

**function FileUpload00\_onload(obj:FileUpload, e:FileUploadLoadEventInfo)**

**{**

**if( e.datasets == null )**

**{**

**alert("업로드 실패");**

**} else {**

**alert("업로드 성공");**

**}**

**}**

## 질문유형

* 파일 업로드 실패시에도 true가 리턴 됩니다.

## 기타

* 접수번호 :
* 조회건수 :