Flash Upload

使用说明

|  |  |
| --- | --- |
| 编制人员： | 陈大新 |
| 编制部门： | CCS |

2009年5月

版本记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 修改人 | 修改内容 | 批准人 | 生效日期 | 版本号 |
| 1 | 陈大新 | 新建 |  | 2009/5/6 | 1.0 |
| 2 | 陈大新 | 修改了2.1章节JS参数的说明 |  | 2009/5/13 | 1.1 |
| 3 | 王珏 | 修改js的构造方法，以及参数的组织 |  | 2009/6/10 | 1.2 |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |

目 录

[1. 概述 3](#_Toc229372125)

[2. 使用说明 4](#_Toc229372126)

[2.1. 页面引用 4](#_Toc229372127)

[2.2. 服务端处理 4](#_Toc229372128)

[3. 条件与限制 6](#_Toc229372129)

[4. 附：FlashUpload Bug 7](#_Toc229372130)

# 概述

FlashUpload是一个使用Flash技术上传文件的一个组件，该组件最大的好处就是上传的进度实时可见，并且使用简单，不需要传统方法在页面上编写大量的Javascript。



图 FlashUpload界面

# 使用说明

## 页面引用

在页面加载Flash Upload控件需要引用/js/wiscom.js文件，再需要显示上传控件的地方调用方法：

var flashVars = WE.util.flashUploadVars;

flashVars.fileCount = fileCount;

flashVars.fileSizeLimit = fileSizeLimit\*1024\*1024+"";

flashVars.totalUploadSize=totalUploadSize\*1024\*1024;

WE.util.FlashUpload (url, params, flashVars)

参数说明如下：

|  |  |
| --- | --- |
| **参数** | **说明** |
| url | 接收上传文件的URL，一般来说是一个servlet或一个controller |
| params | 额外参数 |
| flashVars | 文件上传的限定条件集合（map形式）包括 上传文件的个数、最大上传大小、单个文件的最大上传大小、文件类型的过滤、上传成功后的回调函数等设置，具体如下  jsessionid : 用于在非IE浏览器上传文件时保持session fileSizeLimit :单个文件大小的限制，0为不限制，字节. totalUploadSize :文件总大小的限制，0为不限制，字节. fileCount : 总文件的个数，0为不限制.  fileTypes : 文件类型，多个类型用半角分号分隔开fileTypeDescription :文件类型描述  completeFunction :文件上传完成后调用页面Javascript函数的函数名 |

其中除url外其它参数都是可选参数。

控件的宽X高为：590X380。

一个典型的使用方法为：

<script type="text/javascript" src="${contextPath}/js/wiscom.js"></script>

<table width="590">

<tr>

<td height="380">

<script type="text/javascript">

var params = {'nodeId':'123', 'groupName':'CCS项目组'};

var flashVars = WE.util.flashUploadVars; flashVars.fileTypes = "\*.jpg;\*.gif"; flashVars.fileTypeDescription = "图片文件（.jpg;\*.gif）";

flashVars.fileCount = "5";

flashVars.fileSizeLimit = "1024000";

flashVars.onUploadComplete=”doit”;

var flashUpload = new WE.util.FlashUpload("/homework/upload.do", params, flashVars );

flashUpload.show();

// 或者：flashUpload.show(‘someDiv’);

// 或者：document.write(flashUpload);

</script>

</td>

</tr>

</table>

<div id=”someDiv”></div>

本例中在表格中显示了一个Flash Upload控件，并设定上传的文件路径为：/homework/upload.do，添加了两个额外的参数，nodeId和groupName，文件全部上传完成后调用本页面的Javascript函数onUploadComplete，最多上传10个文件，总文件大小限制为10240000字节，设置 上传文件的类型“flashVars.fileTypes = "\*.jpg;\*.gif"”。 如果不输入限制则没有限制。

由此例可以看出，显示一个FlashUpload控件分为两步：

1.声明FlashUpload对象

2.设置 FlashUpload的 限定条件WE.util.flashUploadVars

3.显示FlashUpload控件

其中显示FlashUpload控件的方法有：

1. 调用show(容器ID)方法
2. 通过toString()方法获取FlashUpload的HTML，使用document.write()方法显示

## 服务端处理

服务端处理有两种方法，使用Controller中的FileWebInput内置的方法，和使用FlashUploadWrapper工具类，推荐使用前者。

### 使用FileWebInput

使用FileWebInput只需要在Controller对应的方法中声明一个类型为FileWebInput的参数，下面显示了一个典型用法。

/\*\*

\* 上传一个文件.

\*

\* @param webin FileWebInput类，封装了常用的上传文件的处理方法.

\* @return Null.

\*/

public ModelAndView uploadFile(final FileWebInput webin) throws Exception {

// 这里的file name是指上传的文件域的名称，而不是文件名称

String[] fileNames = webin.getFileNames();

if (fileNames != null && fileNames.length > 0) {

// 获取一个MultipartFile，这是spring的一个工具类

MultipartFile file = webin.getMultipartFile(fileNames[0]);

// 上传文件的文件名

String originalName = file.getOriginalFilename();

// 保存目标文件

File destFile = new File("/D:/Temp/" + originalName);

file.transferTo(destFile);

logger.debug("Get file name({}) from request, and save to {}.", originalName, destFile.getAbsolutePath());

}

return null;

}

### 使用FlashUploadWrapper

Flash Upload客户端可以一次性选择多个文件，但上传的时候是每个文件发起一次HTTP请求，因此服务端每次只能接受到一个文件。

在协作平台中可以使用工具类：

com.wiscom.ccs.utils.FlashUploadWrapper

处理上传的文件。

FlashUploadWrapper类只能在使用Apache Common File Upload的Spring MVC容器使用，其它容器由于上传文件的机制不一样因此不能使用。

本类使用时需要传递HttpServletRequest作为构造函数，构造成功后就可以使用该类的方法，方法说明如下：

/\*\*

\* 获取上传文件的原文件名.

\*

\* @return 上传文件的原文件名.

\*/

public String getFileName();

/\*\*

\* 将文件保存到目标位置.

\*

\* @param destFile 目标位置.

\* @throws IOException 写文件出错.

\*/

public void saveTo(File destFile) throws IOException;

/\*\*

\* 是否上传了文件.

\*

\* @return 是否上传了文件.

\*/

public boolean hasFile();

# 条件与限制

在非IE浏览器中，Flash不能将本地的cookie发送到接受上传文件的Http地址，即在WE.util.flashUpload（）方法中的第一个参数，如果该地址采用的是cookie认证，则认证无法通过，因为浏览器的cookie没有一并发送过去，因此接受上传文件的servlet需要重新编写认证模块，并且不能将该URL加入到统一的认证过滤器中。

该Flash Bug详细说明请参见：[附：Flash Upload Bug](#_附：FlashUpload_Bug)。

## IDS认证Cookie问题解决方案

在CCS系统中已经添加了一个变通的解决方案来解决取不到cookie的问题。系统中的com.wiscom.ccs.service.impl.IDSLoginServiceImpl如果不能从客户端获取cookie会尝试从request的参数中去获取cookie值，因此Flash Upload Bug在CCS中的解决方法就是把cookie值写到request的参数中。

具体做法分为两步：

* 将cookie值在controller中取出，返回给页面
* 在页面的JS中添加相应的参数

在controller中取得cookie值可以参考以下代码：

HashMap params = new HashMap();

params.put(...); // 设置其它值

// 如果是用ids认证，则需要添加特殊的变量

if (loginService.getClass().isAssignableFrom(IDSLoginServiceImpl.class)) {

IDSLoginServiceImpl idsLoginService = (IDSLoginServiceImpl) loginService;

params.put(idsLoginService.getTokenCookieName(),

idsLoginService.getToken(webin.getRequest()));

}

return new ModelAndView("some.view", params);

在页面生成FlashUpload控件时加上一个参数：

<script type="text/javascript">

var params = {

'param1' : "value1", // 程序需要作用的值

'iPlanetDirectoryPro' : '${iPlanetDirectoryPro}' // cookie值

};

WE.util.flashUpload("${contextPath}/some/controller.do", params);

</script>

# 附：Flash Upload Bug

From: <http://www.swfupload.org/forum/generaldiscussion/383>

I want to clarify what I have observed about the Flash Cookie bug.

The Flash Player Plugin for FireFox, Opera and Safari (and probably other non-IE based browsers) has a bug which sends persistent cookies from IE to the upload URL instead of the cookies from the browser. Session only cookies from IE are not sent.

When Flash initializes in the browser its own empty "cookie space" is created. It loads persistent cookies from IE (which you can see in %USERPROFILE%\cookies). In-memory (session) cookies are not loaded.

The cookies from the browser are not loaded in to Flash's cookie space.

Any session cookies created by the upload script are maintained in-memory in Flash's cookies space. New persistant cookies are created on disk (which you can see in %USERPROFILE%\cookies) and will immediately appear in IE. Cookies created in the Flash cookie space will not appear in any of the browser's "view cookie" tools.

All Flash Movies share the same per browser cookie space which is maintained until the browser is closed (i.e., multiple tabs in FireFox will share the same Flash cookie space but FireFox and Safari maintain separate Flash cookie spaces).

I've carefully tested this issue in FireFox 3 and IE 7 on Windows XP Pro with Flash Player 9.0.115. I also did some basic testing in Opera 9.24 and the Safari Beta for Windows. I plan to create a new demo which will demonstrate my findings.

I have not tested this issue on OS X or in Linux.

―――――――――――<全文结束>――――――――――