

Uploadify 中文 API

Uploadify 使用要求

jQuery 的 1.4.x 或更高

Flash 播放器 9.0.24 或更高

Uploadify 实现步骤

1、下载 [Uploadify zip 包](#)。

2、解压缩包，将以下必要文件拷贝到项目 webapp 下的文件中：

- [jquery.uploadify 3.1.min.js](#)
- uploadify.php
- [uploadify.swf](#)
- [uploadify.css](#)
- [uploadify cancel.png](#)
- check-exists.php

红色标注的为 java 必要。

3、在网页中引入 jQuery 库。

```
<script type="text/javascript" src="http://code.jquery.com/jquery-1.7.2.min.js"></script>
```

4、在 jQuery 库下面引入 uploadify.min.js。

```
<script type="text/javascript" src="/jquery.uploadify-3.1.min.js"></script>
```

5、在页面上加入一个 input 标签。

```
<input type="file" name="file_upload" id="file_upload" />
```

6、初始化 uploadify。

```
$(function() {  
  
    $('#file_upload').uploadify({  
  
        'swf' : 'uploadify.swf',  
  
        'uploader' : 'uploadify.php'  
  
        // Put your options here  
  
    });  
  
});
```

7、增加 uploadify 的样式文件

```
<link rel="stylesheet" type="text/css" href="uploadify.css" />
```

Demo.jsp 如下图:

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <title>My Uploadify Implementation</title>  
5     <link rel="stylesheet" type="text/css" href="uploadify.css">  
6     <script type="text/javascript" src="http://code.jquery.com/jquery-1.7.2.min.js"></script>  
7     <script type="text/javascript" src="jquery.uploadify-3.1.min.js"></script>  
8     <script type="text/javascript">  
9         $(function() {  
10             $('#file_upload').uploadify({  
11                 'swf' : 'uploadify.swf',  
12                 'uploader' : 'uploadify.php'  
13                 // Your options here  
14             });  
15         });  
16     </script>  
17 </head>  
18 <body>  
19 <input type="file" name="file_upload" id="file_upload" />  
20 </body>  
21 </html>
```

注: java 中需要将图中用荧光笔画出的地方替换为: 'uploader': 'actionName!methodName.do'

Demo.java (上传文件相关)

private File upload; // 文件

private String uploadFileName; // 文件名

```
public void upLoadImg() throws Exception {

    FileInputStream ins = new FileInputStream(upload);// 将上传文件转为输入流

    String uri =                                     //文件保存的uri

    FileOutputStream out = new FileOutputStream(uri);

    byte[] buf = new byte[1024];

    for (int readNum; (readNum = ins.read(buf)) != -1;) {

        out.write(buf, 0, readNum);

    }

    out.close();

    ins.close();

}


public File getUpload() {

    return upload;

}


public void setUpload(File upload) {

    this.upload = upload;

}


public String getUploadFileName() {
```

```
        return uploadFileName;
    }

    public void setUploadFileName(String uploadFileName) {

        this.uploadFileName = uploadFileName;
    }
}
```

Uploadify 的选项

optionName	explain	default value
auto	表示在选择文件后是否自动上传	TRUE
buttonClass	额外增加的上传按钮样式类型	
buttonCursor	上传按钮 Hover 时的鼠标形状	hand
buttonImage	按钮的背景图片	
buttonText	按钮的背景图片	
checkExisting	若要检查可指明一个用于判断的脚本的路径	FALSE
debug	开启 DEBUG 模	FALSE
fileObjName	文件对象名称。用于在服务器端获取文件	Filedata

fileSizeLimit	上传文件大小限制，默认单位是 KB	
fileTypeDesc	文件类型的说明	
fileTypeExts	指定允许上传的文件类型(文件后缀名)	*, *
formData	指定上传文件附带的其他数据(json 格式)	
height	表示按钮的高度	30px
itemTemplate	允许将 html 模版加入队列中	FALSE
method	上传方式(post or get)	post
multi	是否支持多文件上传	TRUE
overrideEvents	设置需要覆写的事件名称，格式为['', '']	
preventCaching	一个随机数将被加载 swf 文件 URL 的后面，防止浏览器缓存	TRUE
progressData	设置文件上传时显示的数据，有两个选择：‘上传速度’或者‘百分比’，分别对应‘speed’和‘percentage’	
queueID	上传队列的 DOM 元素的 ID 号，设置了 queueID，那么选择多个文件上传后，就能看到实时的队列情况	
queueSizeLimit	队列长度限制	999

<code>removeCompleted</code>	在上传完成后是否删除队列中的对应元素，即上传完成后就看不到上传文件进度条了。	TRUE
<code>removeTimeout</code>	上传完成后多久删除队列中的进度条	3 秒
<code>requeueErrors</code>	设置为 true，上传过程中因为出错导致上传失败的文件将被重新加入队列	FALSE
<code>successTimeout</code>	文件上传完成后等待服务器响应的时间。超过该时间，那么将认为上传成功	30 秒
<code>swf</code>	swf 文件路径	
<code>uploader</code>	php 为服务器端脚本文件路径，java 为方法路径	
<code>uploadLimit</code>	最多上传文件数量	999
<code>width</code>	按钮宽度	120

Uploadify 的事件

事件调用规则：

```
'event':function(parameter){

}
```

不同的事件有不同的参数。

- `onCancel`

说明：

在文件被移除出队列时触发

示例：

```
<a href="javascript:$('#file_upload').uploadify('upload')">Upload Files</a>
```

```
'onCancel' : function(file) {  
  
    alert('The file ' + file.name + ' was cancelled.');
```

```
}  
}
```

参数：

file: 被取消的文件对象

- [onClearQueue](#)

说明：

在调用 cancel 方法且传入参数 '*' 时触发

示例：

```
<a href="javascript:$('#file_upload').uploadify('cancel','*');">Clear Queue</a>
```

```
'onClearQueue' : function(queueItemCount) {  
  
    alert(queueItemCount + ' file(s) were removed from the queue');
```

```
}  
}
```

参数：

queueItemCount: 队列中项目的数量

- [onDestroy](#)

说明：

调用 destroy 方法时触发

示例：

```
<a href="javascript:$('#file_upload').uploadify('destroy')">Destroy Uploadify</a>
```

```
'onDestroy' : function() {  
  
    alert('I am getting destroyed!');  
  
}
```

- **onDialogClose**

说明:

打开文件对话框关闭时触发

示例:

```
'onDialogClose' : function(queueData) {  
  
}
```

QueueData 包括:

filesSelected: 浏览文件时选择的文件的数量

filesQueued: 添加到队列中的文件的数量

filesReplaced: 替换队列中的文件数

filesCancelled: 添加到队队列中被取消的文件数

filesErrored: 返回了错误的文件数

- **onDialogOpen**

说明:

浏览文件对话框打开时触发

示例:

```
'onDialogOpen' : function() {  
  
}
```


- [onDisable](#)

说明：

当 uploadify 被禁用时触发

示例

```
'onDisable' : function() {  
  
    alert('You have disabled Uploadify!');  
  
}
```

- [onEnable](#)

说明：

当 uploadify 可以使用时触发

示例：

```
'onEnable' : function() {  
  
    alert('You can use Uploadify again.');
```

```
}  

```

- [onFallback](#)

说明：

没有兼容的 flash 时触发

示例：

```
'onFallback' : function() {  
  
    alert('Flash was not detected.');
```

```
}  

```

- [onInit](#)

说明：

Uploadify 第一次被调用时触发

示例：

```
'onInit' : function(instance) {  
  
    alert('The queue ID is ' + instance.settings.queueID);  
  
}
```

参数：

Instance: uploadify 对象的实例

- [onQueueComplete](#)

说明：

已处理队列中所有的文件时触发

示例：

```
'onQueueComplete' : function(queueData) {  
  
    alert(queueData.uploadsSuccessful + ' files were successfully uploaded.');
```

```
}  
}
```

参数：

queueData 包含：

uploadsSuccessful: 上传成功的数量

uploadsErrored: 上传返回错误的数量

- [onSelect](#)

说明：

在选择的一个文件被添加到队列前触发

示例：

```
'onSelect' : function(file) {  
  
    alert('The file ' + file.name + ' was added to the queue.');
```

```
}  
}
```

参数：

file：被选中的文件对象

- [onSelectError](#)

说明：

将选择文件后每一个返回的错误消息，集中在一个错误提示框中

示例：

```
'onSelectError' : function() {  
  
    alert('The file ' + file.name + ' returned an error and was not added to the queue.');
```

```
}  
}
```

参数：

file：返回错误消息的文件对象

errorCode：

QUEUE_LIMIT_EXCEEDED –选择的文件数量超过了队列的 size

FILE_EXCEEDS_SIZE_LIMIT –文件大小超过限制

ZERO_BYTE_FILE –文件没有大小

INVALID_FILETYPE –文件类型不匹配

errorMsg： 错误消息

- [onSWFReady](#)

说明：

加载 flash 对象时触发

示例：

```
'onSWFReady': function() {  
  
    alert('The Flash file is ready to go.');  
}
```

- [onUploadComplete](#)

说明：

每个文件上传完成时触发，不论是成功还是失败

示例：

```
'onUploadComplete': function(file) {  
  
    alert('The file ' + file.name + ' finished processing.');  
}
```

参数：

file: 文件对象

- [onUploadError](#)

说明：

当一个文件已经上传，但是返回一个错误时触发

示例：

```
'onUploadError': function(file, errorCode, errorMsg, errorString) {  
  
    alert('The file ' + file.name + ' could not be uploaded: ' + errorString);  
  
}
```

参数:

file: 被上传的文件对象

errorCode: 返回的错误码

errorMsg: 返回的错误消息

errorString: 包含错误细节的消息

- [onUploadProgress](#)

说明:

每次上传文件更新进度时触发

示例:

```
'onUploadProgress': function(file, bytesUploaded, bytesTotal, totalBytesUploaded,
    totalBytesTotal) {

    $('#progress').html(totalBytesUploaded + ' bytes uploaded of ' + totalBytesTotal
        + ' bytes.');
```

```
}
```

参数:

file: 被上传的文件对象

bytesUploaded: 文件已经被上传的 byte 数

bytesTotal: 文件总的 byte 数

totalBytesUploaded: 当前上传操作中总的 byte 数(所有文件)

totalBytesTotal: 上传的总字节数 (所有文件)

- [onUploadStart](#)

说明:

上传文件前触发

示例：

```
'onUploadStart' : function(file) {  
  
    alert('Starting to upload ' + file.name);  
  
}
```

参数：

file: 被上传的文件对象

- [onUploadSuccess](#)

说明：

每个文件上传成功后触发

示例：

```
'onUploadSuccess' : function(file, data, response) {  
  
    alert('The file ' + file.name + ' was successfully uploaded with a response of ' +  
        response + ':' + data);  
  
}
```

参数：

file: 被成功上传的文件对象

data: 服务器端返回的数据

response: 由服务器返回的响应。true / false。如 successTimeout 选项到期后，返回 true

Uploadify 的方法

- [cancel](#)

说明：从队列或正在进行的文件中取消一个文件

示例：

```
<a href="javascript:${'#file_upload'}.uploadify('cancel')">Cancel First File</a>
```

```
<a href="javascript:${'#file_upload'}.uploadify('cancel', '*')">Clear the Queue</a>
```

```
<a href="javascript:${'#file_upload'}.uploadify('upload', '*')">Upload the Files</a>
```

参数:

fileId: 要取消的文件的 id, 若没有参数则取消队列的第一个文件, 若参数为* 则取消队列中的所有文件

如果 `suppressEvent` 被设置为 `true`, `onUploadCancel` 事件将被抑制, 常用于清除队列

- [destroy](#)

说明:

销毁 `uploadify` 实例文件的输入, 并返回原始状态

示例:

```
<a href="javascript:${'#file_upload'}.uploadify('destroy')">Destroy Uploadify</a>
```

- [disable](#)

说明:

禁用或启用浏览文件的按钮

示例:

```
<a href="javascript:${'#file_upload'}.uploadify('disable',true)">Disable Button</a>
```

```
<a href="javascript:${'#file_upload'}.uploadify('disable', false)">Enable Button</a>
```

参数:

setDisabled: 设置为 `true` 则禁用按钮, 为 `false` 则启用按钮

- [settings](#)

说明:

返回或更新 `uploadify` 实例中的设置值

示例:

```
<a href="javascript:changeBtnText()">Change the Button Text</a>
```

```
function changeBtnText() {  
  
    $('#file_upload').uploadify('settings','buttonText','BROWSE');  
  
}
```

```
<a href="javascript:returnBtnText();">Read the Button</a>
```

```
function returnBtnText() {  
  
    alert('The button says ' + $('#file_upload').uploadify('settings','buttonText'));  
  
}
```

参数:

name: 你想返回的设置或更改的设置的名称，仅设置此参数将只返回值

value: 更改设置的值

resetObjects: 设置为 `true`，若目标设置已有值，则删除有的值；

设置为 `false`，否则将新值替换现有值

注：不能设置 `swf`

- [stop](#)

说明:

停止当前和所有队列的上传

示例:

```
<a href="javascript:$('#file_upload').uploadify('upload','*')">Upload the Files</a>
```

```
<a href="javascript:$('#file_upload').uploadify('stop')">Stop the Uploads!</a>
```


- [upload](#)

说明：

上传特定的文件或队列中的所有文件

示例：

```
<a href="javascript:$('#file_upload').uploadify('upload','*')">Upload Files</a>
```

参数：

fileId：当 fileId 为文件 id 时，指定该 id 的文件进行上传

当 fileId 为*时，所有文件将按现有的队列进行上传

Uploadify 配置示例

```
$("#uploadfile").uploadify({
    'auto':true,
    'buttonClass':'btn3',
    'buttonText':'批量上传',
    'height':30,
    'fileTypeDesc': 'Image Files',
    'cancellImg': 'images/cancel.png',
    'fileTypeExts': '*.gif; *.jpg; *.png; *.bmp; *.jpeg',
    'fileSizeLimit': '2048KB',
    'queueSizeLimit': 10,
    'fileObjName':'storeImage',
    'swf': '<%=ctxPath%>/ta/resource/themes/base/uploadify/uploadify.swf',
    'uploader': 'sellerStoreAction!uploadStoreImages.do',
    'onUploadSuccess': function(file, data, response) {
    },
    'onQueueComplete': function(queueData) {
    }
});
```