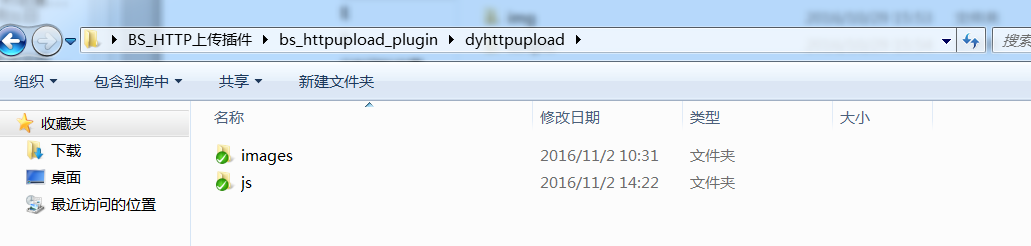
HTTP上传BS插件使用说明文档

**文档变更纪录**

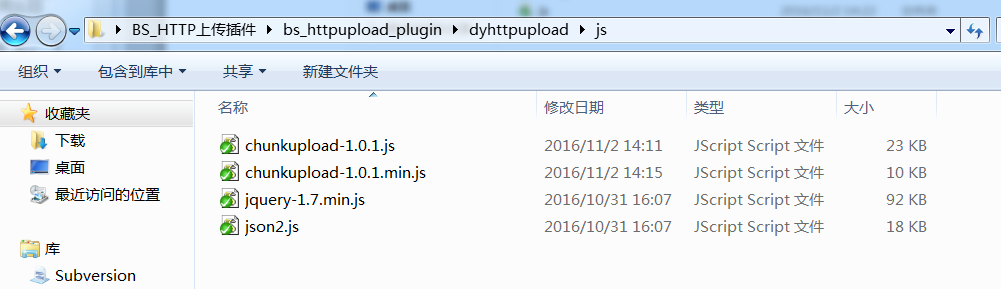
|  |  |  |
| --- | --- | --- |
| **更改人** | **日期** | **更改内容** |
| 任育伟 | 2016.11.2 | 创建文件 |
|  |  |  |

# 1 安装步骤

1. 将插件包下的img和scripts文件夹拷贝至工程目录，两个文件夹名称不能改动且必须在同一级目录。



1. 然后在使用插件页面引用js文件夹下的js文件。插件js代码chunkupload-1.0.1.js和chunkupload-1.0.1.min.js分别是源码和压缩后的代码，两者都可以引用使用。



1. 最后使用jquery选择器渲染插件页面，可以在调用httpUploadPlugin方法时传入参数，代码如下所示，选择器的id是使用页面嵌入插件的标签ID

|  |
| --- |
| <div id="httpupoad">  </div>  <script type="text/javascript" charset="utf-8" src="dyhttpupload/js/jquery-1.7.min.js"></script>  <script type="text/javascript" charset="utf-8" src="dyhttpupload/js/json2.js"></script>  <script type="text/javascript" charset="utf-8" src="dyhttpupload/js/chunkupload-1.0.1.min.js">  </script>  <script type="text/javascript">  var plugin = $("#httpupoad").httpUploadPlugin({  "uploadAddress": "http://100.0.1.211:80/upload/",  "retryMigrateAddress": "http://100.0.1.211:8080/DYCMigrate/rs/retryMigrate",  "chunkSize": 10 \* 1024 \* 1024,  "fileSavePath": "/tmp/upload\_set/123"  });  <script> |

1. 安装成功后，刷新使用插件的页面，出现如下所示页面表示安装成功



1. 点击上传按钮，选择上传文件后，出现如下的上传界面。



# 2 初始化参数

使用插件时，可以在httpUploadPlugin方法传入JSON格式的参数，参数要求具体如下表。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性名称** | **名称** | **必填** | **类型** | **说明** |
| uploadAddress | 上传文件地址 | Y | string | 上传文件地址，必填项，格式如：http://ip:port/upload/ |
| retryMigrateAddress | 重试迁移任务地址 | Y | string | 迁移重试任务地址，必填项,格式如：http://ip:port/ DYCMigrate/rs/ retryMigrate |
| chunkSize | 分片大小 | N | int | 分片大小，以字节为单位，不传则默认为5M |
| fileSavePath | 文件迁移后的保存路径 | Y | string | 迁移后的文件保存路径，必填项,格式如：/tmp/upload\_tmp |
| isRename | 是否重命名 | N | boolean | 是否重命名，true重命名；false不重命名；不传则默认不重命名 |

默认的参数如下：

{ "uploadAddress": "",

"retryMigrateAddress": "",

"chunkSize": 5 \* 1024 \* 1024,

"isRename":false,

"fileSavePath":""

}

使用时传入的参数如下：

{

"uploadAddress": "http://100.0.1.211:80/upload/",

"retryMigrateAddress": "http://100.0.1.211:8080/DYCMigrate/rs/retryMigrate",

"chunkSize": 10 \* 1024 \* 1024,

"fileSavePath": "/tmp/upload\_set/123",

"isRename":true

}

# 3 对外提供方法

使用该var plugin = $("#httpupoad").httpUploadPlugin({

"uploadAddress": "http://100.0.1.211:80/upload/",

"retryMigrateAddress": "http://100.0.1.211:8080/DYCMigrate/rs/retryMigrate",

"chunkSize": 10 \* 1024 \* 1024,

"fileSavePath": "/tmp/upload\_set/123"

});语句获取插件对象后，可以调用插件的isFinishedAllTasks和getUploadTasksDetail两个方法,，方法介绍如3.1和3.2。

## 3.1 isFinishedAllTasks(是否成功上传所有任务)

**描述**：是否成功上传所有任务，true表示所有任务上传完成，false表示不是所有任务上传完成。

**输入参数**

**无**

**输出参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性名称** | **名称** | **必填** | **类型** | **说明** |
| finishStatus | 完成状态 | Y | boolean | true都上传成功,false,不是所有任务上传完成 |
| description | 描述 | N | string | 结果描述 |

**参数样例**

|  |
| --- |
| {  " finishStatus ": true,  " description ": "发生异常"  } |

## 3.2 getUploadTasksDetail(获取上传任务详情)

**描述**：获取所有上传任务的上传结果详情，通过JSON格式数据返回。

**输入参数**

**无**

**输出参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **属性名称** | **名称** | **必填** | **类型** | **说明** |
| uploadTasksDetail | 上传任务详情 | Y | boolean | 上传任务详情 |
| ---fileSessionId | 文件sessionId | Y | boolean | 文件sessionId |
| ---fileName | 原文件名称 | Y | string | 上传文件名称 |
| ---migrateFileName | 迁移后文件名称 | Y | string | 迁移后的文件名称 |
| --- stop | 是否暂停上传 | Y | boolean | true，表示暂停  false,表示正在上传 |
| ---offset | 文件下片的上传位置 | Y | boolean | 文件的偏移位置 |
| ---finishProgress | 文件上传进度 | Y | boolean | 0-100,100表示上传完成 |
| ---responseStatus | 返回的响应状态码 | Y | boolean | 状态码 |
| description | 描述 | N | string | 结果描述 |

**参数样例**

|  |
| --- |
| {  " uploadTasksDetail ": [{  " fileSessionId": "0123456",  " fileName ": " filename.mp4 ",  " migrateFileName": " filename.mp4 ",  " offset": "4",  " stop": true,  " finishProgress": "0",  " responseStatus": "200"  },...]  " description ": "发生异常"  } |

更多信息可以参考《HTTP上传插件开发文档》