E:\work\公司VI\公司logo(曲线）.wmf

**前端插件——批量上传**

**（v0.1）**

项目部/前端架构组

2010年 12月

版本

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 作者 | 评审人 | 修订内容 | 版本 |
| 2010-12-6 | 李丽媛 | 丁镇涛 |  | 0.1 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 概要设计 1](#_Toc280862687)

[1.1. 需求概要 1](#_Toc280862688)

[1.2. 功能需求 1](#_Toc280862689)

[1.2.1. 选择上传文件（见图【1】） 2](#_Toc280862690)

[1.2.2. 上传文件的文件名（见图【1】） 2](#_Toc280862691)

[1.2.3. 文件大小（见图【1】） 2](#_Toc280862692)

[1.2.4. 文件上传状态（见图【1】） 2](#_Toc280862693)

[1.2.5. 取消上传/删除已上传或正在上传的文件（见图【1】） 2](#_Toc280862694)

[1.2.6. 上传进度（见图【1】） 2](#_Toc280862695)

[1.2.7. 上传文件（见图【1】） 2](#_Toc280862696)

[1.2.8. 取消上传（见图【1】） 2](#_Toc280862697)

[1.2.9. 完成上传，关闭上传面板（见图【1】） 2](#_Toc280862698)

[1.2.10. 打开上传面板（见图【1】） 3](#_Toc280862699)

[2. 详细设计 4](#_Toc280862700)

[2.1. 功能描述 4](#_Toc280862701)

[2.2. Init 4](#_Toc280862702)

[2.3. Options 4](#_Toc280862703)

[2.4. Events 5](#_Toc280862704)

[3. 工作开展方式与进度计划 9](#_Toc280862705)

[3.1. 开展方式 9](#_Toc280862706)

[3.2. 进度计划 9](#_Toc280862707)

# 概要设计

## 需求概要

针对目前现有系统中的上传功能，它们都存在不易使用、难以修改、无法维护等特点。因此需结合各个系统中的实际使用情况对上传这一功能进行分析与改造。进而得出一个通用、简洁、便于开发和维护的上传插件。经分析，该插件应具备以下主要功能：

* 显示单个文件的上传进度、文件大小及上传状态
* 删除已经上传的文件
* 限定上传文件的大小及类型
* 批量上传
* 取文件上传（正在上传或等待上传的文件）
* 自定义样式

## 功能需求

根据分析，上传插件应具备以下功能，其主要结构可参见图【1】：

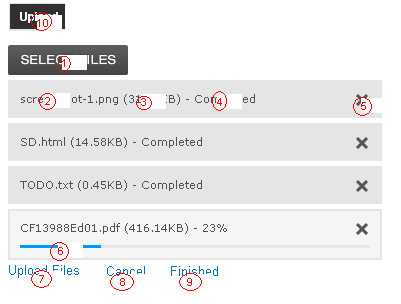


图 1

### 选择上传文件（见图【1】）

点击按钮，弹出选择文件窗口，选择需要上传的文件。

### 上传文件的文件名（见图【1】）

对[1.2.1](#_选择上传文件（见图【1】）)中选中的文件，应显示其文件名称。

### 文件大小（见图【1】）

对[1.2.1](#_选择上传文件（见图【1】）)中选中的文件，应显示其文件大小。

### 文件上传状态（见图【1】）

显示每一个文件的当前状态：已上传、正在上传、上传中。

### 取消上传/删除已上传或正在上传的文件（见图【1】）

对正在上传的文件、已经上传的文件、还未进行上传的文件可以进行删除。

### 上传进度（见图【1】）

按照百分比显示单个文件的上传进度。

### 上传文件（见图【1】）

点击按钮，对[1.2.1](#_选择上传文件（见图【1】）)中选中的文件进行全部上传。

### 取消上传（见图【1】）

点击按钮，取消当前操作，关闭上传文件的操作面板。

### 完成上传，关闭上传面板（见图【1】）

在所有文件上传完毕后，点击按钮，关闭上传文件的操作面板。

### 打开上传面板（见图【1】）

点击按钮，显示上传文件的操作面板，可在其中对上传文件进行相关操作。

# 详细设计

## 功能描述

* 批量上传
* 上传文件类型限制
* 上传文件大小限制
* 上传文件框中文件类型的提示信息
* 设置上传到服务器的路径
* 设置最大可上传文件的数目
* 自定义
* 上传按钮的图片和文字
* 取消文件按钮的图片
* 根据 CSS 自定义样式
* 加载完毕，上传完成，出现错误，取消上传时可触发用户自定义事件

## Init

.upload();

## Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 类型 | 默认值 | 说明 | 示例 |
| buttonImg | String | 无 | 按钮图片。 | 'buttonImg': '/uploadify/button.png' |
| buttonText | String | 无 | 按钮上的文字。 | ‘buttonText’: ‘upload’ |
| cancelImg | String | 无 | 取消文件的图片。 | 'cancelImg': '/uploadify/cancel.png' |
| fileExt | String | 无 | 可上传的文件类型。 | 'fileExt': '\*.jpg;\*.gif;\*.png' |
| fileDesc | String | 无 | 选择上传文件框中，文件类型的提示。 | 'fileDesc': 'Web Image Files (.JPG, .GIF, .PNG)' |
| folder | String | 无 | 上传到服务器的路径。 | 'folder': '/uploads' |
| multi | Boolean | false | 允许上传多个文件。 | 'multi': true |
| queueSizeLimit | Int | 999 | 面板中可显示文件的数目。 | 'queueSizeLimit' : 3 |
| sizeLimit' | Int | 无 | 限制每个上传文件的最大值。 | 'sizeLimit': 102400 |

## Events

|  |  |  |
| --- | --- | --- |
| 事件 | 参数 | 说明 |
| ****onSelectOnce(event, data)**** | event: event 对象  data: {  fileCount: 文件数目,  filesSelected: 被选中的文件数目,  filesReplaced: 被替换的文件数目,  allBytesTotal: 文件总大小  } | 每一次添加文件时，触发该事件。 |
| onSelect(event, id, fileObj) | event: event 对象  id: 被选中文件的唯一标示  fileObj: {  name: 文件名称,  size: 文件大小,  creationDate: 文件创建日期,  modificationData: 文件修改日期,  type:文件类型  } | 每天加一个文件时，触发该事件。 |
| onAllComplete(event, data) | event: event 对象  data: {  filesUpload: 上传文件数目,  errors: 发生错误的次数,  allBytesLoaded: 上传总大小,  speed: 上传的平均速度  } | 所有文件上传完后，触发该事件。 |
| onCancel(event, id, fileObj, data) | event: event 对象  id: 被取消文件的唯一标示  fileObj: {  name: 文件名称,  size: 文件大小,  creationDate: 文件创建日期,  modificationData: 文件修改日期,  type:文件类型  }  data: {  fileCount: 剩下的文件数目,  allBytesTotle: 剩下文件的总大小  } | 取消一个文件时，触发该事件。 |
| onClearQueue(event) | event: event 对象 | 移除所有文件时，触发该事件。 |
| onError | id: 上传文件返回错误的唯一标示  fileObj: {  name: 文件名称,  size: 文件大小,  creationDate: 文件创建日期,  modificationData: 文件修改日期,  type:文件类型  }  errorObj: {  type: ‘HTTP’ / ‘IO’ / ‘Security’,  info: 错误信息  } | 当上传错误时，触发该事件。 |
| onInit() | 无 | 该插件加载完时，触发该事件。 |
| onComplete(event, id, fileObj, response, data) | event: event 对象  id: 已上传文件的唯一标示  fileObj: {  name: 文件名称,  filePath: 文件位于服务器的地址,  size: 文件大小,  creationDate: 文件创建日期,  modificationData: 文件修改日期,  type:文件类型  }  response: 返回信息  data: {  fileCount: 剩下的文件数目,  speed: 平均上传速度  } | 每一个文件上传完时，触发改事件。 |
| onCheck(event, data, key) | event: event 对象  data: 服务器上已存在的文件名称列表  key: 在服务其上检测到重复文件的唯一标示 | 当文件名重复时，触发该事件。 |

# 工作开展方式与进度计划

## 开展方式

* 目前，此类插件已经非常完善，完全可以满足现有系统中的文件上传功能。因此，决定从第三方插件中进行选取，不再进行自主开发；
* 结合各个系统的实际情况及第三方插件的功能特点进行开展：
* 首先，对上传的第三方插件进行研究，从中选取一款较为合适的插件;
* 其次，按照[详细设计](#_详细设计)对插件进行封装;
* 最后，按照[概要设计](#_概要设计)及[主要功能](#_主要功能)对插件进行相应的修改。

## 进度计划

在没有其他任务分配及前置条件满足的前提下，该插件开发进度如下：

2010-12-23 至 2010-12-27（3个工作日）

* 根据需求选取第三方插件；
* 熟悉插件代码，根据[概要设计](#_概要设计)及[详细设计](#_详细设计)对其进行封装及修改；
* 与底层进行集成。

2010-12-28 至 2010-12-29（2个工作日，需在界面设计完成的前提下）

* 根据界面设计，做出相应皮肤的 Demo；
* 内部测试及 bug 修改。