引入资源

使用 Web Uploader 文件上传需要引入三种资源：JS, CSS, SWF。

```
<!--引入 CSS--><link rel="stylesheet" type="text/css" href="webuploader 文件夹/webuploader.css">

<!--引入 JS--><script type="text/javascript" src="webuploader 文件夹/webuploader.js"></script>

<!--SWF 在初始化的时候指定，在后面将展示-->
```

文件上传

WebUploader 只包含文件上传的底层实现，不包括 UI 部分。所以交互方面可以自由发挥，以下将演示如何去实现一个简单的版本。

请点击下面的选择文件按钮，然后点击开始上传体验此 Demo。

选择文件

开始上传

Html 部分

首先准备 dom 结构，包含存放文件信息的容器、选择按钮和上传按钮三个部分。

```
<div id="uploader" class="wu-example">
    <!--用来存放文件信息-->
    <div id="thelist" class="uploader-list"></div>
    <div class="btns">
        <div id="picker">选择文件</div>
        <button id="ctlBtn" class="btn btn-default">开始上传</button>
    </div></div>
```

初始化 Web Uploader

具体说明，请看一下代码中的注释部分。

```
var uploader = WebUploader.create({
```

```
    // swf 文件路径
    swf: BASE_URL + '/js/Uploader.swf',

    // 文件接收服务端。
    server: 'http://webuploader.duapp.com/server/fileupload.php',

    // 选择文件的按钮。可选。
    // 内部根据当前运行是创建，可能是 input 元素，也可能是 flash.
    pick: '#picker',

    // 不压缩 image, 默认如果是 jpeg，文件上传前会压缩一把再上传！
    resize: false});
```

显示用户选择

由于 webuploader 不处理 UI 逻辑，所以需要去监听 `fileQueued` 事件来实现。

```
// 当有文件被添加进队列的时候uploader.on( 'fileQueued', function( file ) {
    $list.append( '<div id="' + file.id + '" class="item">' +
        '<h4 class="info">' + file.name + '</h4>' +
        '<p class="state">等待上传...</p>' +
    '</div>' );});
```

文件上传进度

文件上传中，Web Uploader 会对外派送 `uploadProgress` 事件，其中包含文件对象和该文件当前上传进度。

```
// 文件上传过程中创建进度条实时显示。uploader.on( 'uploadProgress', function( file, percentage ) {
    var $li = $( '#'+file.id ),
        $percent = $li.find('.progress .progress-bar');
```

```
    // 避免重复创建
    if ( !$percent.length ) {
        $percent = $('<div class="progress progress-striped active">' +
          '<div class="progress-bar" role="progressbar" style="width: 0%">' +
          '</div>' +
        '</div>').appendTo( $li ).find('.progress-bar');
    }

    $li.find('p.state').text('上传中');

    $percent.css( 'width', percentage * 100 + '%' );});
```

文件成功、失败处理

文件上传失败会派送 `uploadError` 事件，成功则派送 `uploadSuccess` 事件。不管成功或者失败，在文件上传完后都会触发 `uploadComplete` 事件。

```
uploader.on( 'uploadSuccess', function( file ) {
    $( '#'+file.id ).find('p.state').text('已上传');});
uploader.on( 'uploadError', function( file ) {
    $( '#'+file.id ).find('p.state').text('上传出错');});
uploader.on( 'uploadComplete', function( file ) {
    $( '#'+file.id ).find('.progress').fadeOut();});
```

图片上传

与普通文件上传相比，此 demo 演示了：文件过滤，图片预览，图片压缩上传等功能。

选择图片

Html

要实现如上 demo，首先需要准备一个按钮，和一个用来存放添加的文件信息列表的容器。

```
<!--dom 结构部分--><div id="uploader-demo">
    <!--用来存放 item-->
    <div id="fileList" class="uploader-list"></div>
    <div id="filePicker">选择图片</div></div>
```

Javascript

创建 Web Uploader 实例

```
// 初始化 Web Uploadervar uploader = WebUploader.create({

    // 选完文件后，是否自动上传。
    auto: true,

    // swf 文件路径
    swf: BASE_URL + '/js/Uploader.swf',

    // 文件接收服务端。
    server: 'http://webuploader.duapp.com/server/fileupload.php',

    // 选择文件的按钮。可选。
    // 内部根据当前运行是创建，可能是 input 元素，也可能是 flash.
    pick: '#filePicker',

    // 只允许选择图片文件。
    accept: {
        title: 'Images',
        extensions: 'gif,jpg,jpeg,bmp,png',
        mimeTypes: 'image/*'
```

```
});
```

监听 `fileQueued` 事件，通过 `uploader.makeThumb` 来创建图片预览图。
PS：这里得到的是 Data URL 数据，IE6、IE7 不支持直接预览。可以借助 FLASH 或者服务端来完成预览。

```
// 当有文件添加进来的时候uploader.on( 'fileQueued', function( file ) {
    var $li = $(
            '<div id="' + file.id + '" class="file-item thumbnail">' +
                '<img>' +
                '<div class="info">' + file.name + '</div>' +
            '</div>'
            ),
        $img = $li.find('img');



    // $list 为容器 jQuery 实例
    $list.append( $li );


    // 创建缩略图
    // 如果为非图片文件，可以不用调用此方法。
    // thumbnailWidth x thumbnailHeight 为 100 x 100
    uploader.makeThumb( file, function( error, src ) {
        if ( error ) {
            $img.replaceWith('<span>不能预览</span>');
            return;
        }


        $img.attr( 'src', src );
    }, thumbnailWidth, thumbnailHeight );});
```

然后剩下的就是上传状态提示了，当文件上传过程中，上传成功，上传失败，上传完成都分别对应 uploadProgress, uploadSuccess, uploadError, uploadComplete 事件。

```
// 文件上传过程中创建进度条实时显示。uploader.on( 'uploadProgress', function( file, percentage ) {
    var $li = $( '#'+file.id ),
        $percent = $li.find('.progress span');

    // 避免重复创建
    if ( !$percent.length ) {
        $percent = $('<p class="progress"><span></span></p>')
                .appendTo( $li )
                .find('span');
    }

    $percent.css( 'width', percentage * 100 + '%' );});
// 文件上传成功，给 item 添加成功 class, 用样式标记上传成功。uploader.on( 'uploadSuccess', function( file ) {
    $( '#'+file.id ).addClass('upload-state-done');});
// 文件上传失败，显示上传出错。uploader.on( 'uploadError', function( file ) {
    var $li = $( '#'+file.id ),
        $error = $li.find('div.error');

    // 避免重复创建
    if ( !$error.length ) {
        $error = $('<div class="error"></div>').appendTo( $li );
    }

    $error.text('上传失败');});
// 完成上传完了，成功或者失败，先删除进度条。uploader.on( 'uploadComplete', function( file ) {
```

```
    $( '#'+file.id ).find('.progress').remove();});
```