File 在 ajax 进行提交的时候 提交的时候 后端返回的数据格式

{"name":"adorn\_bg.png","type":"image\/png","tmp\_name":"E:\\Xampp\\tmp\\php2F09.tmp","error":0,"size":2667}

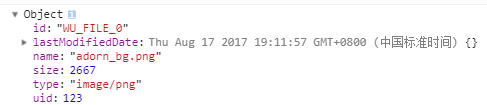
|  |  |
| --- | --- |
|  | Array |
|  | ( |
|  | [name] => adorn\_bg.png |
|  | [type] => image/png |
|  | [tmp\_name] => E:\Xampp\tmp\php4CC5.tmp |
|  | [error] => 0 |
|  | [size] => 2667 |
|  | ) |

Webuplaoer

File 的对象的信息如下



发送给后端的数据格式如下：



多个文件夹的时候是分开传递的，不是一下子传递一个数据上去

uploader.getStats() 返回的数据信息：



Onerror 返回的函数

uploader.onError = function( code ) {}

Code 返回的值

F\_EXCEED\_SIZE :验证单个文件大小是否超出限制, 超出则不允许加入队列

F\_DUPLICATE:去重， 根据文件名字、文件大小和最后修改时间来生成hash Key

Q\_TYPE\_DENIED:类型不匹配，则派送错误事件，并返回

Q\_EXCEED\_NUM\_LIMIT:上传文件总个数超过规定的个数

\* \* `Q\_EXCEED\_NUM\_LIMIT` 在设置了`fileNumLimit`且尝试给`uploader`添加的文件数量超出这个值时派送。

\* \* `Q\_EXCEED\_SIZE\_LIMIT` 在设置了`Q\_EXCEED\_SIZE\_LIMIT`且尝试给`uploader`添加的文件总大小超出这个值时派送。

\* \* `Q\_TYPE\_DENIED` 当文件类型不满足时触发。。

file.on('statuschange', function( cur, prev ){

'statuschange' 可以获取文件的状态

File.Status

/\*\*

\* 获取文件状态

\* @return {File.Status}

\* @example

文件状态具体包括以下几种类型：

{

// 初始化

INITED: 0,

// 已入队列

QUEUED: 1,

// 正在上传

PROGRESS: 2,

// 上传出错

ERROR: 3,

// 上传成功

COMPLETE: 4,

// 上传取消

CANCELLED: 5

}

\*/

getStatus()

file.on('statuschange', function( cur, prev ) {

/\*\*

\* 文件状态值，具体包括以下几种类型：

\* \* `inited` 初始状态

\* \* `queued` 已经进入队列, 等待上传

\* \* `progress` 上传中

\* \* `complete` 上传完成。

\* \* `error` 上传出错，可重试

\* \* `interrupt` 上传中断，可续传。

\* \* `invalid` 文件不合格，不能重试上传。会自动从队列中移除。

\* \* `cancelled` 文件被移除。

getStats():



uploader.refresh();

http://www.jianshu.com/p/b59ebac54c43

uploader.upload();

调用这个方法以后，就会触发下面一系列的事件：

startUpload

uploadStart

uploadBeforeSend

uploadProgress

uploadAccept

uploadSuccess

uploadComplete

uploadFinished 所有的文件上传结束

