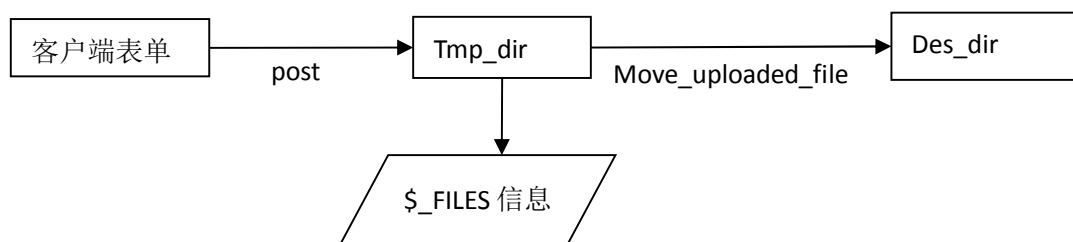


第 15 讲 文件的上传操作

林世鑫 2017 年 9 月

在动态网站中，文件上传是比较常见的功能之一。PHP 为用户提供了非常方便的文件上传操作，实现的代码也比较简单。

使用 PHP 的文件上传功能时，通常要在配置文件 php.ini 中先对上传作一些必须的设置，然后通过 PHP 的预定义变量\$_FILES 获取文件的一些属性信息，并对其进行合法性判断，最后利用 PHP 的文件上传函数 move_uploaded_file() 将上传的文件移到指定目录中，即可实现文件的上传。该流程可以归纳如下图：



15.1、配置 php.ini 文件

PHP.ini 文件中，有一些默认的设置，对文件的上传操作，会有很大的影响，为了保证文件上传的顺利进行，在进行文件上传操作之前，通常要先对这些设置根据实际情况，作出一定的修改。它涉及到的主要内容项有如下（通过 phpstudy—>其它选项菜单—>打开配置文件）：

(1) file_uploads:该项的值如果为 on，表示服务器支持上传，如果为 off，则不支持；见下图：

```
880 ; whether to allow HTTP file uploads.
881 ; http://php.net/file-uploads
882 file_uploads = on
```

(2) `upload_tmp_dir`: 该项用于指定文件上传的临时目录, 在文件上传成功之前, 文件首先存放在该临时目录中, 用户可以另外指定一个目录, 如果未指定, 则使用系统的默认目录。见下图:

```
884 ; Temporary directory for HTTP uploaded files (will use system default if not
885 ; specified).
886 ; http://php.net/upload-tmp-dir
887 ; upload_tmp_dir =
```

(3) `upload_max_filesize`: 允许上传的最大文件值, 单位是 MB。系统默认是 2M, 用户可以自定义大小。见下图:

```
889 ; Maximum allowed size for uploaded files.
890 ; http://php.net/upload-max-filesize
891 upload_max_filesize = 2M
```

(4) `max_execution_time`: 该选项用于指定一个 PHP 指令的最长执行时间, 单位是秒。它的值, 对文件上传的影响主要在于: 如果文件太大, 上传时间过长, 就会超过了指令的最长执行时间, 这时指令会停止执行, 从而导致文件上传失败。因此, 通常对于大文件的上传, 也相应需要把该项的值设置大一些。见下图:

```
441 ; Maximum execution time of each script, in seconds
442 ; http://php.net/max-execution-time
443 ; Note: This directive is hardcoded to 0 for the CLI SAPI
444 max_execution_time = 30
```

(5) `memory_limit`: 用于设置 PHP 一个指令所分配的内存空间, 单位是 MB。如果上传的文件体积较大, 该项配置的值通常也需要调整大一些, 以便造成上传时间过长, 脚本执行超时。见下图:

```
463 ; Maximum amount of memory a script may consume (128MB)
464 ; http://php.net/memory-limit
465 memory_limit = 128M
```

(6) `max_file_uploads`: 一次性所能上传的最多文件数。默认值是 20 个。用户可以自定义。见下图:

```
893 ; Maximum number of files that can be uploaded via a single request
894 max_file_uploads = 20
```

以上配置选项，在 phpstudy 中，通常都是已经按常见情况，默认配置好了。只有在特殊的情况下，我们才需要另外再配置。

15.2、预定义变量\$_FILES

\$_FILES 变量是一个二维数组，它的元素中，保存了文件上传时的一系列属性信息，并且用不同的属性名作为元素的键名。因此，通过这些键名对不同的元素进行访问，可以得到文件上传操作中，文件的各种属性值。它的主要元素列表及内容如下：

(1) \$_FILES['myFile']['name']：保存的是客户端上传文件原名。

(2) \$_FILES['myFile']['type']：文件的类型。需要注意的是，该项内容与浏览器环境紧密关联，它与 windows 下的文件类型是有区别的，同样的 windows 文件类型，在不同浏览器环境下，PHP 返回的类型值并不一样。例如“jpg”格式的文件，在 IE 浏览器中，返回的类型是“image/jpeg”，而在火狐浏览器中即是“image/pjpeg”。（更多类型详情，参阅附录）

(3) \$_FILES['myFile']['size']：已上传文件的大小，单位为字节。

(4) \$_FILES['myFile']['tmp_name']：因为 PHP 默认是先将文件上传到服务器端的临时文件夹中的（PHP.ini 文件的 upload_tmp_dir 设置），因此它也有一个临时的文件名，该名称则保存在“tmp_name”元素中。

(5) \$_FILES['myFile']['error']：文件上传相关的错误代码。文件上传的过程中，如果发生了错误，会伴随着错误返回一个错误码，保存在“error”元素中，通过该错误码的值，可以判断发生了什么错误。下

表是不同错误码所代表的错误内容：

错误名	码值	内容含义
UPLOAD_ERR_OK	0	没有错误发生，文件上传成功。
UPLOAD_ERR_INI_SIZE	1	上传的文件超过了 php.ini 中 upload_max_filesize 选项限制的值
UPLOAD_ERR_FORM_SIZE	2	上传文件的大小超过了 MAX_FILE_SIZE 选项指定的值
UPLOAD_ERR_PARTIAL	3	文件只有部分被上传
UPLOAD_ERR_NO_FILE	4	没有文件被上传
	5	上传文件大小为 0

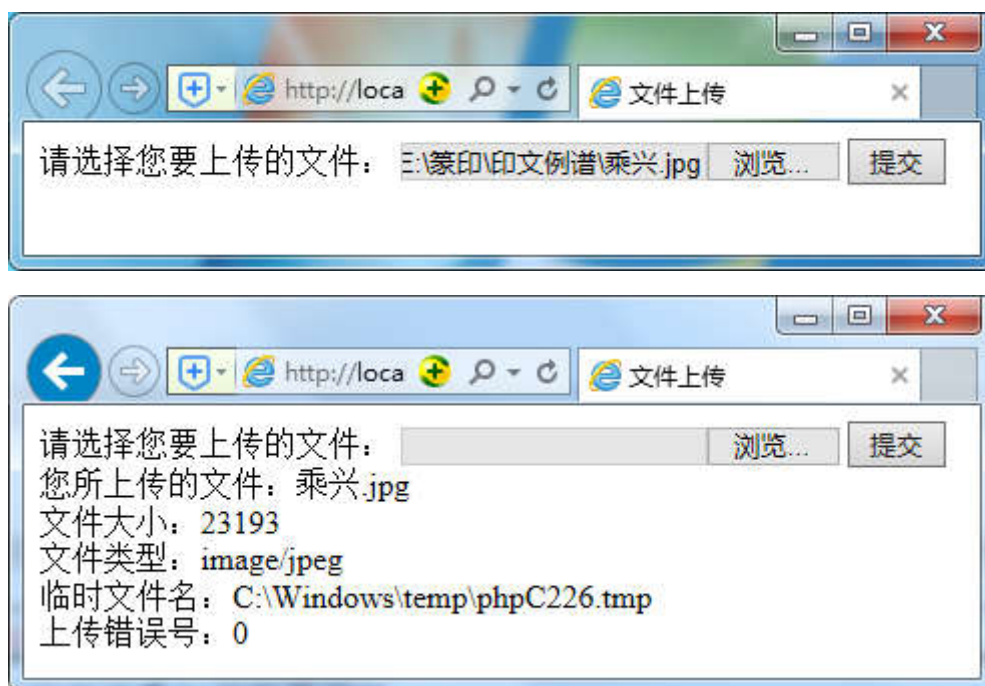
例如在一个表单中，进行文件上传提交，并返回文件上传的结果：

【例 15-2】

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>文件上传</title>
</head>
<body>
<form action="" method="post" enctype="multipart/form-data"
name="form1" id="form1">
    请选择您要上传的文件：
    <label for="myfile"></label>
    <input type="file" name="myfile" id="myfile" />
    <input type="submit" name="button" id="button" value="提交" />
</form>
<?php
    if(!empty($_FILES))
    {
        echo "您所上传的文件：".$_FILES['myfile']['name']."<br>";
        echo "文件大小：".$_FILES['myfile']['size']."<br>";
        echo "文件类型：".$_FILES['myfile']['type']."<br>";
        echo "临时文件名：".$_FILES['myfile']['tmp_name']."<br>";
        echo "上传错误号：".$_FILES['myfile']['error'];
    }
?>
</body>
```

```
</html>
```

运行的结果如下两图所示：



从上例可以看出，windows 下，服务器默认的临时文件夹是“C:\windows\temp\”，文件提交后，文件就暂时保存在这个文件夹中。

用 PHP 进行文件上传操作时，有几点需要注意：

(1) 文件被上传结束后，只是暂时存储在临时目录中，必须将它从临时目录中移动到服务器的其它文件夹，才能完成真正的上传，否则脚本执行完后临时目录里的文件肯定会被删除。

(2) 用表单上传文件时，一定要设置属性“编码类型”的内容为“multipart/form-data”，否则用\$_FILES[filename]获取文件信息时会报异常。

(3) 使用 post 方法上传表单数据，以保证文件安全上传。

15.3、move_uploaded_file()函数

PHP 的 move_uploaded_file() 用于将上传在临时目录中的文件，移动

到其它的目录下，从而完成文件的上传过程。其语法格式如下：

```
move_uploaded_file(文件名, 目标路径)
```

其中，“文件名”是保存在临时目录中的临时文件名，可以通过
\$_FILES 数组中的[tmp_name]获得；

“目标路径”用于指定文件保存的新目录以及文件名。

文件移动成功后，函数返回 true，否则返回 false。

例如利用表单，上传一个大小不超过 1MB、类型为.jpg 或.gif 格式的
图像文件。

【15-3】

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=gb2312" />
<title>图片上传</title>
</head>
<body>
<form action="" method="post" enctype="multipart/form-data"
name="form1" id="form1">
    <table width="369" height="72" border="0" align="center"
cellpadding="0" cellspacing="0">
        <tr>
            <td height="36">图片上传（请勿大于 1M）： </td>
        </tr>
        <tr>
            <td height="36"><label for="mypic"></label>
            &nbsp;&nbsp;&nbsp;<input type="file" name="mypic" id="mypic" />
            <input type="submit" name="button" id="button" value="上传
" /></td>
        </tr>
    </table>
</form>
<?php
    if(!empty($_FILES['mypic']['name'])) //判断是否上传图片
    {
```

```
$f_type=$_FILES['mypic']['type'];
$f_size=$_FILES['mypic']['size'];
$temp_name=$_FILES['mypic']['tmp_name'];
$c_dir=getcwd();//获取当前文件夹作为上传目录
//判断文件类型

if($f_type!='image/gif'&&$f_type!='image/jpeg'&&$f_type!='image/pjpeg')
    {echo "文件类型不对，请重新上传". "<br>";
    exit;}
//判断文件大小
if($f_size>1024*1024)
    {echo "文件超出许可大小，请重新上传<br>";
    exit;}
$n_name=$c_dir."\\".$_FILES['mypic']['name'];
$upic=move_uploaded_file($temp_name,$n_name);
if($upic)
    {echo "图像上传成功<br>";}
else
    {echo "图像上传失败";}
}
?>
</body>
</html>
```

【附录】PHP 对 windows 文件格式的识别类型

后缀名	php 识别出的文件类型（火狐）	php 识别出的文件类型（IE）
gif	image/gif	image/gif
jpg	image/pjpeg	image/jpeg
png	image/x-png	image/png
bmp	image/bmp	image/bmp
psd	application/octet-stream	application/octet-stream
ico	image/x-icon	image/x-icon
rar	application/octet-stream	application/octet-stream
zip	application/x-zip-compressed	application/zip
7z	application/octet-stream	application/octet-stream
exe	application/octet-stream	application/octet-stream
avi	video/avi	video/avi

后缀名	php 识别出的文件类型（火狐）	php 识别出的文件类型（IE）
rmvb	application/vnd.rn-realmedia-vbr	application/vnd.rn-realmedia-vbr
3gp	application/octet-stream	application/octet-stream
flv	application/octet-stream	application/octet-stream
mp3	audio/mpeg	audio/mpeg
wav	audio/wav	audio/wav
krc	application/octet-stream	application/octet-stream
lrc	application/octet-stream	application/octet-stream
txt	text/plain	text/plain
doc	application/msword	application/msword
xls	application/vnd.ms-excel	application/vnd.ms-excel
ppt	application/vnd.ms-powerpoint	application/vnd.ms-powerpoint
pdf	application/pdf	application/pdf
chm	application/octet-stream	application/octet-stream
mdb	application/msaccess	application/msaccess
sql	text/plain	application/octet-stream
con	application/octet-stream	application/octet-stream
log	text/plain	text/plain
dat	text/plain	application/octet-stream
ini	application/octet-stream	application/octet-stream
php	application/octet-stream	application/octet-stream
html	text/html	text/html
htm	text/html	text/html
ttf	application/octet-stream	application/octet-stream
fon	application/octet-stream	application/octet-stream
js	text/html	application/x-javascript
xml	text/xml	text/xml
dll	application/octet-stream	application/octet-stream
dll	application/java	application/octet-stream

15.4、思考与练习

请设计实现一个多文件上传页面，允许用户动态增加、删除要上传的

文件数，所有文件不得大于 1M，文件类型不限，全部上传到“files”目录下，上传以后每个文件以上传“时间戳 N”命名，其中 N 为文件在该批次上传列表中的序号，从 0 开始。（提示：参考第 9 讲数组（二））。