**POST 方法上传**

本特性可以使用户上传文本和二进制文件。用 PHP 的认证和文件操作函数，可以完全控制允许哪些人上传以及文件上传后怎样处理。

PHP 能够接受任何来自符合 RFC-1867 标准的浏览器（包括 Netscape Navigator 3 及更高版本，打了补丁的 Microsoft Internet Explorer 3 或者更高版本）上传的文件。

**相关的设置:**请参阅 php.ini 的 [file\_uploads](http://www.jb51.net/shouce/php5/zh/ini.core.html#ini.file-uploads)，[upload\_max\_filesize](http://www.jb51.net/shouce/php5/zh/ini.core.html#ini.upload-max-filesize)，[upload\_tmp\_dir](http://www.jb51.net/shouce/php5/zh/ini.core.html#ini.upload-tmp-dir)[post\_max\_size](http://www.jb51.net/shouce/php5/zh/ini.core.html#ini.post-max-size) 以及 [max\_input\_time](http://www.jb51.net/shouce/php5/zh/ref.info.html#ini.max-input-time) 设置选项。

请注意 PHP 也支持 PUT 方法的文件上传，Netscape Composer 和 W3C 的 Amaya 客户端使用这种方法。请参阅[对 PUT 方法的支持](http://www.jb51.net/shouce/php5/zh/features.file-upload.put-method.html)以获取更多信息。

|  |  |
| --- | --- |
| **例子 38-1. 文件上传表单**  可以如下建立一个特殊的表单来支持文件上传：   |  | | --- | | <!-- The data encoding type, enctype, MUST be specified as below -->  <form enctype="multipart/form-data" action="\_\_URL\_\_" method="POST">  <!-- MAX\_FILE\_SIZE must precede the file input field -->  <input type="hidden" name="MAX\_FILE\_SIZE" value="30000" />  <!-- Name of input element determines name in $\_FILES array -->  Send this file: <input name="userfile" type="file" />  <input type="submit" value="Send File" />  </form> |   以上范例中的 *\_\_URL\_\_* 应该被换掉，指向一个真实的 PHP 文件。  *MAX\_FILE\_SIZE* 隐藏字段（单位为字节）必须放在文件输入字段之前，其值为接收文件的最大尺寸。这是对浏览器的一个建议，PHP 也会检查此项。在浏览器端可以简单绕过此设置，因此不要指望用此特性来阻挡大文件。实际上，PHP 设置中的上传文件最大值是不会失效的。但是最好还是在表单中加上此项目，因为它可以避免用户在花时间等待上传大文件之后才发现文件过大上传失败的麻烦。 |

**注:**要确保文件上传表单的属性是 *enctype="multipart/form-data"*，否则文件上传不了。

全局变量 [$\_FILES](http://www.jb51.net/shouce/php5/zh/reserved.variables.html#reserved.variables.files) 自 PHP 4.1.0 起存在（在更早的版本中用 *$HTTP\_POST\_FILES* 替代）。此数组包含有所有上传的文件信息。

以上范例中 [$\_FILES](http://www.jb51.net/shouce/php5/zh/reserved.variables.html#reserved.variables.files) 数组的内容如下所示。我们假设文件上传字段的名称如上例所示，为 *userfile*。名称可随意命名。

*$\_FILES['userfile']['name']*

客户端机器文件的原名称。

*$\_FILES['userfile']['type']*

文件的 MIME 类型，如果浏览器提供此信息的话。一个例子是“*image/gif*”。不过此 MIME 类型在 PHP 端并不检查，因此不要想当然认为有这个值。

*$\_FILES['userfile']['size']*

已上传文件的大小，单位为字节。

*$\_FILES['userfile']['tmp\_name']*

文件被上传后在服务端储存的临时文件名。

*$\_FILES['userfile']['error']*

和该文件上传相关的[错误代码](http://www.jb51.net/shouce/php5/zh/features.file-upload.errors.html)。此项目是在 PHP 4.2.0 版本中增加的。

文件被上传后，默认地会被储存到服务端的默认临时目录中，除非 php.ini 中的 [upload\_tmp\_dir](http://www.jb51.net/shouce/php5/zh/ini.core.html#ini.upload-tmp-dir) 设置为其它的路径。服务端的默认临时目录可以通过更改 PHP 运行环境的环境变量 *TMPDIR* 来重新设置，但是在 PHP 脚本内部通过运行 [**putenv()**](http://www.jb51.net/shouce/php5/zh/function.putenv.html) 函数来设置是不起作用的。该环境变量也可以用来确认其它的操作也是在上传的文件上进行的。

|  |  |
| --- | --- |
| **例子 38-2. 使文件上传生效**  请查阅函数 [**is\_uploaded\_file()**](http://www.jb51.net/shouce/php5/zh/function.is-uploaded-file.html) 和 [**move\_uploaded\_file()**](http://www.jb51.net/shouce/php5/zh/function.move-uploaded-file.html) 以获取进一步的信息。以下范例处理由表单提供的文件上传。   |  | | --- | | <?php // In PHP versions earlier than 4.1.0, $HTTP\_POST\_FILES should be used instead // of $\_FILES.  $uploaddir = '/var/www/uploads/'; $uploadfile = $uploaddir . basename($\_FILES['userfile']['name']);  echo '<pre>'; if (move\_uploaded\_file($\_FILES['userfile']['tmp\_name'], $uploadfile)) {     echo "File is valid, and was successfully uploaded.\n"; } else {     echo "Possible file upload attack!\n"; }  echo 'Here is some more debugging info:'; print\_r($\_FILES);  print "</pre>";  ?> | |

接受上传文件的 PHP 脚本为了决定接下来要对该文件进行哪些操作，应该实现任何逻辑上必要的检查。例如可以用 *$\_FILES['userfile']['size']* 变量来排除过大或过小的文件，也可以通过 *$\_FILES['userfile']['type']* 变量来排除文件类型和某种标准不相符合的文件，但只把这个当作一系列检查中的第一步，因为此值完全由客户端控制而在 PHP 端并不检查。自 PHP 4.2.0 起，还可以通过 *$\_FILES['userfile']['error']* 变量来根据不同的[错误代码](http://www.jb51.net/shouce/php5/zh/features.file-upload.errors.html)来计划下一步如何处理。不管怎样，要么将该文件从临时目录中删除，要么将其移动到其它的地方。

如果表单中没有选择上传的文件，则 PHP 变量 *$\_FILES['userfile']['size']* 的值将为 0，*$\_FILES['userfile']['tmp\_name']* 将为空。

如果该文件没有被移动到其它地方也没有被改名，则该文件将在表单请求结束时被删除。

|  |  |  |
| --- | --- | --- |
| **例子 38-3. 上传一组文件**  PHP 的 [HTML 数组特性](http://www.jb51.net/shouce/php5/zh/faq.html.html#faq.html.arrays)甚至支持文件类型。   |  | | --- | | <form action="" method="post" enctype="multipart/form-data">  <p>Pictures:  <input type="file" name="pictures[]" />  <input type="file" name="pictures[]" />  <input type="file" name="pictures[]" />  <input type="submit" value="Send" />  </p>  </form> |  |  | | --- | | <?php foreach ($\_FILES["pictures"]["error"] as $key => $error) {     if ($error == UPLOAD\_ERR\_OK) {         $tmp\_name = $\_FILES["pictures"]["tmp\_name"][$key];         $name = $\_FILES["pictures"]["name"][$key];         move\_uploaded\_file($tmp\_name, "data/$name");     } } ?> | |