PHP实现图片的上传与下载

# 开发环境

1、环境搭建：Windows 7+Apache 2.4.18+MySQL 5.7.11+PHP 7.1.0。

2、文本编辑器：Sublime 3。

# 二、主要技术

本案例主要使用PHP 7中文件操作函数来实现图片的上传和下载功能。

# 三、效果图展示

实现效果图，如图1所示。

图1 效果图（图片上传和下载操作界面）

# 四、具体步骤

## 4.1、制作图片信息展示页

### （1）创建php文件和图片文件夹

①创建一个名为“index.php”的PHP文件用来制作图片信息的展示页面。

②在根目录下创建一个名为“upload”的文件夹，用来存储上传的图片。

### （2）界面设计

<form action="doupload.php" method="post" enctype="multipart/form-data">

//最大上传图片大小

<input type="hidden" name="MAX\_FILE\_SIZE" value="100000">

上传图片：<input type="file" name="pic"/> //选择要上传的文件

<input type="submit" name="sub" value="上传"> //提交按钮

</form>

### （3）实现图片信息展示功能

<table width="500" bgcolor="#E6E8FA">

<tr align="left" bgcolor="#CCCCFF">

<th>序号</th>

<th>图片</th>

<th>添加时间</th>

<th>操作</th>

</tr>

<?php

$dir=opendir("./upload");//打开upload目录

//遍历upload目录，输出里面的图片信息

$i=0;

while($f=readdir($dir)){

if($f!="."&&$f!=".."){

$i++;

echo "<tr>";

echo "<td>{$i}</td>";

echo "<td><img src='./upload/{$f}'

width='80' height='50'></td>";

echo "<td>".date("Y-

m- d",filectime("./upload/".$f))."</td>";

echo "<td><a href='./upload/{$f}'>查看</a>&nbsp;

<a href='dodownload.php?name={$f}'>下载</a></td>";

echo "</tr>";

}

}

closedir($dir);//关闭目录

?>

<tr bgcolor="#CCCCFF">

<td colspan="4">&nbsp;</td>

</tr>

</table>

## 4.2、实现图片上传功能

### （1）创建php文件

创建一个名为“doupload.php”的文件，用来实现图片上传的功能。

### （2）获取上传文件信息

$upfile = $\_FILES["pic"];

//定义允许的类型

$typelist = array("image/jpeg","image/jpg","image/png",

"image/gif");

$path="./upload/"; //定义一个上传过后的目录

### （3）过滤上传文件的错误号

if($upfile["error"]>0){

//获取错误信息

switch($upfile['error']){

case 1:

$info="上传的文件超过了 php.ini

中 upload\_max\_filesize 选项限制的值。";

break;

case 2:

$info="上传文件的大小超过了 HTML 表单

中 MAX\_FILE\_SIZE 选项指定的值。";

break;

case 3:

$info="文件只有部分被上传。";

break;

case 4:

$info="没有文件被上传。 ";

//没有错误号5

case 6:

$info="找不到临时文件夹。";

break;

case 7:

$info="文件写入失败";

break;

}

die("上传文件错误，原因：".$info);

}

### （4）过滤上传文件大小

if($upfile["size"]>100000){

die("上传文件大小超出限制！");

}

### （5）过滤上传文件类型

if(!in\_array($upfile["type"],$typelist)){

die("上传文件类型非法！".$upfile["type"]);

}

### （6）定义文件上传后的文件名

//上传后的文件名定义(随机获取一个文件名（保持后缀名不变）)

$fileinfo = pathinfo($upfile["name"]);//解析上传文件名字

do{

$newfile= date("YmdHis").rand(1000,9999).".

".$fileinfo["extension"];

}while(file\_exists($path.$newfile));

### （7）执行文件上传

//判断是否是一个上传的文件

if(is\_uploaded\_file($upfile["tmp\_name"])){

//执行文件上传（移动上传文件）

if(move\_uploaded\_file($upfile["tmp\_name"],

$path.$newfile)){

echo "文件上传成功！";

echo "<h3><a href='index.php'>浏览</a></h3>";

}else{

die("上传文件失败");

}

}else{

die("不是一个上传文件！");

}

## 4.3、实现图片下载功能

### （1）创建php文件

创建一个名为“dodownload.php”的php文件，用来实现图片下载功能。

### （2）获取要下载的图片名

//获取要下载的图片名（加上路径）

$file = "./upload/".$\_GET["name"];

### （3）重设浏览器响应类型

$info = getimagesize($file);

header("Content-Type:".$info["mime"]);

### （4）执行下载的文件名

header("Content-Disposition:attachment;filename=

".$\_GET["name"]);

### （5）指定文件的大小

header("Content-Length:".filesize($file));

### （6）响应内容

readfile($file);

# 五、网页测试

①开启Apache服务器后，打开浏览器，在浏览器地址栏中输入文件地址，例如：“localhost:8080/pic\_upload/index.php”，可以看到程序运行结果如图2所示。



图2 案例结果图（图片信息展示页）

②点击【浏览】，选择一张图片，然后点击【上传】，可以看到程序运行结果如图3所示。

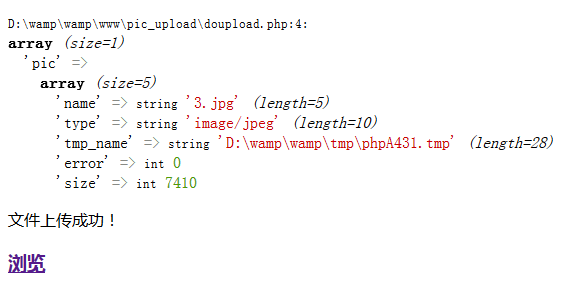


图3 案例结果图（文件上传成功）

③点击“浏览”，可以看到页面跳转到图片信息显示页，并且我们可以看到页面中有了刚刚所上传的图片信息，如图4所示。

图4 案例结果图（图片信息展示页）

④点击刚刚上传图片右侧的“查看”，可以看到程序运行结果如图5所示。

⑤点击刚刚上传图片右侧的“下载”超链接，选择下载到桌面，可以在桌面看到下载到的图片，如图6所示。



图5 案例结果图（查看图片）

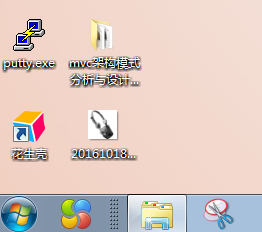


图6 案例结果图（下载到的图片）