

66. 上传功能

学习要点：

1. 上传功能

本节课我们来学习一下系统提供的上传方法，如何使用这个方法进行文件上传。

一. 上传功能

1. 如果要实现上传功能，首先需要建立一个上传表单，具体如下：

```
<form action="http://localhost/tp5.1test3/public/upload"
      enctype="multipart/form-data" method="post">
    <input type="file" name="image">
    <input type="submit" value="确定">
</form>
```

2. 创建一个控制器 `upload.php`，用于处理上传文件；

```
class Upload
{
    public function index()
    {
        //获取表单的上传数据
        $file = Request::file('image');
        //移动到应用目录 uploads 下
        $info = $file->move('../application/uploads');
        //判断上传信息
        if ($info) {
            //输出上传信息
            echo $info->getExtension();
            echo '<br>';
            echo $info->getSaveName();
            echo '<br>';
            echo $info->getFileName();
        } else {
            //输出错误信息
            echo $file->getError();
        }
    }
}
```

3. 批量上传，使用 `image[]` 作为名称，并使用 `foreach()` 遍历上传；

```
<form action="http://localhost/tp5.1test3/public/upload/uploads"
      enctype="multipart/form-data" method="post">
    <input type="file" name="image[]">
```

```
<input type="file" name="image[]">
<input type="file" name="image[]">
<input type="submit" value="确定">
</form>

public function uploads()
{
    //获取表单的上传数据
    $files = Request::file('image');
    foreach ($files as $file) {
        //移动到应用目录 uploads 下
        $info = $file->move('../application/uploads');
        //判断上传信息
        if ($info) {
            //输出上传信息
            echo $info->getExtension();
            echo '<br>';
            echo $info->getSaveName();
            echo '<br>';
            echo $info->getFileName();
        } else {
            //输出错误信息
            echo $file->getError();
        }
    }
}
```

4. 上传文件，可以通过 `validate()` 方法进行验证，包括大小限定、后缀限定等；

```
$info = $file->validate([
    'size' => 102400,
    'ext' => 'jpg,gif,png',
    //'type' => 'text/html'
])->move('../application/uploads');
```

5. 默认情况下，上传的文件是以日期和微秒生成的方式：`date`；

6. 生成的规则还支持另外两种方式：`md5` 和 `sha1`；

```
$file->rule('md5')->move('../application/uploads');
```

7. 也可以通过传递一个方法或函数来自定义命名，比如使用 `uniqid()`；

```
$file->rule('uniqid'); //uniqid()产生一个微秒时间生成一个唯一的 ID
```

8. 在 `move()` 方法的，第二参数设置为空字符串，可以表示按原本的名称保存；

```
$file->move('../application/uploads', '');
```

- 9. 如果设置原本的名称上传，会导致上传同名的文件被新的覆盖；
- 10. 那么也可以设置，在同名的基础上，不去覆盖之前的文件；

```
$file->move('../application/uploads', true, false);
```