

利用laravel框架写留言板。

留言板的功能：

- 1、对留言进行增删改查。
- 2、分页。
- 3、上传图片。
 - 第一次没有上传，修改时没有上传。
 - 第一次没有上传，修改时上传了图片。
 - 第一次上传了，修改时没有上传。
 - 第一次上传了，修改时上传了图片。

具体实现。

- 1、建数据表。（注意表名和模型名是相对应的）
 - 表名：msgform。
- 2、创建模型。
 - 创建模型的命令：
 - php artisan make:model Msgform
 - laravel框架会在app文件夹下自动创建Msgform.php。
 - \$table属性指定表名。即：
 - 默认使用复数形式。但是msgform数据表没有使用复数形式。所以需要在模型类中使用\$table属性来指定表名。

```
class Msgform extends Model
{
    protected $table = 'msgform';
}
```

- 注意：在默认情况下，laravel会自动将模型名转换为表名，**并使用复数形式**，即Msgforms表。（模名转表名，默认是复数！）
- 3、创建控制器：
 - 创建控制器的命令：
 - php artisan make:controller MsgFormController
 - laravel框架会在app/Http/Controllers文件夹下自动创建MsgFormController.php。
- 4、配置路由：
 - 打开routes/web.php。
 - 语法格式：
 - Route::请求方式('请求URI',匿名函数或控制器相应方法);

- 增改:
 - 采集数据


```
Route::get('msgForm/edit/{uid?}', 'MsgFormController@edit');
```

 - 没有uid的时候就是增, 有uid的时候就是改。
 - 保存数据


```
Route::post('msgForm/save', 'MsgFormController@save');
```
- 删:


```
Route::get('msgForm/delete/{uid?}', 'MsgFormController@delete');
```
- 查:


```
Route::get('msgForm/index', 'MsgFormController@index');
```

• 一般来说, **需要实现某种功能的时候就去配置相应的路由。**

• 5、增删改查:

◦ 增改:

- 在MsgFormController编写edit方法。
 - 有没有传uid?
 - 没传: 添加操作。
 - 传了: 更新操作。
 - 取出相应的记录。
 - 查询单行数据。
 - `$data = DB::table('msgform')->where('uid', $uid)->first();`
 - 具体代码。

```
public function edit()
{
    if (isset($_GET['uid'])) {
        $uid = $_GET['uid'];
        // 查询单行数据。
        $data = DB::table('msgform')
            ->where('uid', $uid)->first();
    } else {
        $data = '';
    }
    $data = \compact('data');
    return view('msgForm/edit', $data);
}
```

• 5、增删改查:

◦ 增改:

- 视图:
 - 在resource/views目录中创建msgForm/edit.blade.php
 - 向视图传递的数据是否为空?

```
value="{{empty($data) ? '' : $data->xxx}}"
```

- 具体代码:

```
<form action="save" method="post" enctype="multipart/form-data">
    用户名: <input type="text" name = "username" value="{{empty($data) ? '' : $data->username}}">
    <br/>
    <br/>
    留言内容:
    <textarea name="content" id="" cols="30" rows="10">
        {{empty($data) ? '' : $data->content}}
    </textarea>
    <br>
    @if(!empty($data) && $data->image)
        <div>
            
        </div>
        <input type="hidden" name="lastImg" value="{{ $data->image }}">
    @endif
    <input type="file" name="image" id="">
    <br>
    <input type="hidden" name="_token" value="{{csrf_token()}}">
    <input type="hidden" name="uid" value= "{{empty($data) ? '' : $data->uid}}">
    <input type="submit" value="{{empty($data) ? '留言' : '修改'}}">
</form>
```

- 5、增删改查:

- 查:

- 在MsgFormController编写index方法。

```
public function index()
{
    $collection = DB::table('msgform')->paginate(2);
    $data = \compact('collection');
    return view('msgForm/index', $data);
}
```

- 5、增删改查:

- 查:

- 视图:

- 在resource/views目录中创建msgForm/index.blade.php

- 5、增删改查:

- 删:

- 在MsgFormController编写delete方法。

- 具体代码:

```

public function delete()
{
    // echo '这是删除方法';
    // exit;
    $uid = isset($_GET['uid']) ? $_GET['uid'] : '';
    if (!$uid) {
        return redirect('msgForm/index');
    }
    $re = DB::table('msgform')->where('uid', $uid)->delete();
    if ($re) {
        return redirect('msgForm/index');
    } else {
        echo '删除失败';
    }
}
}

```

- 6、图片上传。
- 7、注意事项：
 - 1、重定向。
 - redirect('msgForm/index');
 - 详情：
 - [重定向](#)
 - 2、图片（静态资源）引入问题。
 - 图片目录：
 - public/images/
 - 引入方法：
 - ``
 - 3、a标签中的href如何写的问题。
 - 在自定义的框架中app类会将URL自动解析为xxx控制器的xxx方法。
 - 例子：
 - `uid}}">更新`
 - `uid}}">删除`
 - **这里不太明白为啥不用写控制器，而只写方法就够了**，如果加了控制器，那么点击更新后显示的路径如下：
 - <http://www.laravel.test/public/index.php/msgForm/msgForm/edit?uid=1>
 - 发现原来就有msgForm，不知道为啥？先不管，毕竟我现在都知道这个问题叫啥。
 - 图片问题。
 - 测试：
 - D:\phpStudy\PHPTutorial\WWW\index\aaa\uploadThump
 - 更新图片：
 - 四种情况：
 - 上一次没有上传，修改时没有上传。

- 上一次没有上传，修改时上传了图片。
- 上一次上传了，修改时没有上传。
- 上一次上传了，修改时上传了图片。
- 压缩：
 - 不管上一次有没有上传，只要修改时上传了图片，就要更新'image'字段。
- 删除图片：
 - 上一次上传了图片，修改时也上传了图片时，就要把上一次上传的图片以及缩略图都删了，不然太占资源了。

note路径： D:\phpStudy\PHPTutorial\WWW\index\aaa\使用laravel\note.md