## Laravel支持多种请求方式

get、put、post等

|  |
| --- |
| <form method=’PUT’ action=’’>  <input type=’hidden’ name=’\_method’ value=’PUT’>//第一种方式  {{ method\_field(‘PUT’) }}//第二种方式  </form> |

## 快速向数据库填充测试数据

使用database/factories/ModelFactory.php

* 效仿例子即可，此处使用了faker，详情：<https://github.com/fzaninotto/Faker>
* return 数组之后，在tinker里面，执行：

|  |
| --- |
| factory（App\Post::class，20）->make()//仅在当前页面显示，不插入进数据库  factory（App\Post::class，20）->create()//插入到数据库  factory（）第二个参数是需要插入多少条记录 |

## 辅助方法（仅仅是laravel提供的）

str\_limit($value, $limit = 100, $end = '...')//限制字符串显示数量，$limit最大显示的字符数，$end 超出显示范围显示的内容

request（）返回当前请求实例,返回一个对象

request()->all() 返回请求传递过来的数组；

request(‘title’) 或者获取一个输入项

request([‘title­’,’content’]) 获取提交过来的两个字段的值

## 依赖注入

定义路由Route::put('posts/{post}','PostController@update');

//红色部分的post应该对应。

在控制器中就可以将模型注入：

|  |
| --- |
| public function update(Request $request,Post $post)  {  //$post可以通过注入，直接使用，它即是通过路由传入的那个id的Post实例,省去了find一步。  if ($post->update(\request()->all())) {  return redirect('/posts/'.$post->id);  } else {  echo '文章更新失败';  }  //若只是用$request,应该如下：  $post = Post::find($request->id);  if ($post->update(\request()->all())) {  return redirect('/posts/'.$post->id);  } else {  echo '文章更新失败';  }  } |

## 查找一个门脸或注入类有哪些方法可以使用

* 1. 使用tinker

app(‘门脸名|注入类’) 返回对应的类文件名

注入类开头不能有反斜杠

* 1. 使用官方api接口文档

<http://d.laravel-china.org/api/5.4/Illuminate/Log/Writer.html>

* 1. 在这类文件里使用phpstrom的structure视图（左侧栏）

## Auth门面

在模型类里引入use Illuminate\Foundation\Auth\User as Authenticatable;

config/auth.php中：

|  |
| --- |
| 'defaults' => [  'guard' => 'web',  'passwords' => 'users', ],  'guards' => [  'web' => [  'driver' => 'session',使用session  'provider' => 'users',  ],  ]  'providers' => [  'users' => [  'driver' => 'eloquent',模型驱动  'model' => App\User::class,验证的模型是User  ],  ] |

应用：

|  |
| --- |
| $userInfo = request(['email','password']); $is\_remember = *boolval*(\request('is\_remember')); if (Auth::attempt($userInfo, $is\_remember)) {  return redirect('posts'); } else {  return redirect()->back()->withErrors('登录失败')->withInput(); }  Auth::attempt()第一个参数，是要验证的邮箱和密码的数组，第二个参数，是否需要记住密码  Auth::id() 获取当前登录用户的id；  Auth::user()获取当前登录用户的对象实例，可以进一步获取Auth::user()->name等属性 |