

Laravel 5.5 入门教程

By IT崖柏图

Mail 973714522@qq.com

出自 布尔教育PHP高端教育培训

19 章 中间件

对于一个Web应用来说, 在一个请求真正处理前, 我们可能会对请求做各种各样的判断, 然后才可以让它继续传递到更深层次中。而如果我们用 if else 这样子来, 一旦需要判断的条件越来越多, 会使得代码更加难以维护, 系统间的耦合会增加, 而中间件就可以解决这个问题。我们可以把这些判断独立出来做成中间件, 可以很方便的过滤请求, 更直接的说, 可以插入到 `XxController@Method()` 方法前或后去执行。以影响执行的结果, 从而让你的代码更加灵活。

19.1 生成中间件

命令: `php artisan make:middleware EmailMiddleware`

生成的中间件位于

`<project>/app/Http/Middleware/EmailMiddleware`

```
class EmailMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        echo '前';
        return $next($request);
    }
}
```

19.2 注册中间件

将中间件注册到路由的匿名函数:

```
Route::get('/midlle', ['middleware' => 'App\Http\Middleware\EmailMiddleware', function() {
    echo 'luyou';
}]);
```

将中间件注册到控制器(官方手册没有明确说明):

```
//路由
Route::get('/mi', ['middleware' => 'App\Http\Middleware\EmailMiddleware', 'uses' => 'IndexController@mi']);

//控制器
class IndexController extends Controller
{
    public function mi(){
        echo 'middle';
    }
}
```

19.3 注册后发邮件

安装 email 类: `composer require nette/mail`

修改路由:

```
Route::post('auth/register',
[
'middleware'=>'App\Http\Middleware>EmailMiddleware',
'uses'=>'Auth\AuthController@postRegister'
]);
```

中间件:

```
// Http/Middleware/EmailMiddleware

use Nette\Mail\Message;
use Nette\Mail\SmtpMailer;
public function handle($request, Closure $next)
{
    $rs = $next($request);
    // 注册后发邮件

    //$request->user()->email;

    if($request->user()) {
        $mail = new Message;
        # 注意 , 下行的 setFrom 要和你的邮箱名保持一致
        $mail->setFrom('laowang <superkeysir@163.com>')
            ->addTo($request->user()->email)
            ->setSubject(' 试试我的中间件 ')
            ->setBody(" 真好用 !");

        $mailer = new SmtpMailer(array(
            'host' => 'smtp.163.com',
            'username' => 'superkeysir',# 你的 163 用户名
            'password' => 'admins'# 你的邮箱密码
        ));

        $mailer->send($mail);
    }

    return $rs;
}
```

中间件也可以注册在控制器上

```
//检测用户是否登陆,系统已将中间件封装好,直接使用即可
Route::middleware('auth')->group(function(){

});
```

由于默认跳转到路由别名上,所以我们需要在login路由上定义命名路由,如下

```
Route::get('login','LoginController@showLoginForm')->name('login');
```

中间件也可以注册在全局上:

```
// 全局路由是在 <project>/App/Http/kernel.php 上指定
/**
 * The application's global HTTP middleware stack.
 *
 * @var array
 */
// 例 : 这是系统注册的几个全局路由
protected $middleware = [
    \Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,
    \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
    \App\Http\Middleware\TrimStrings::class,
```

```
\Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,  
\App\Http\Middleware\TrustProxies::class,  
];
```

20章 在线支付

20.1 在线支付原理

有两种途径

1,商家直接开通一个网上支付的协议,然后通过银行给商家一个商户编号,然后直接调用网上银行的支付接口.(各种大银行,中农工建),但是这种途径一般小公司看不上,只是因为你的资金流动小.

2,商家直接通过第三方平台,然后第三方平台资金比较雄厚,他联系各个银行,把他们的支付口整合到一起我们直接调用第三方平台即可.(支付宝,网银在线,paypal,等很多);

20.2第三方平台支付调用(Chinabank为例)

通过网银在线文档

1,参数:v_amount v_moneytype v_oid v_mid v_url key

2,v_md5info(验证商户,安全),怎么生成的呢?

MD5校验串生成方法:当消费者在商户端生成最终订单的时候,将订单中的v_amount v_moneytype v_oid v_mid v_url key六个参数的``value``值拼成一个无间隔的字符串(顺序不要改变)

```
public function pay(Request $req){  
    $row = [];  
    $row['v_amount'] = sprintf('%.2f',$req->money);  
    $row['v_moneytype'] = 'CNY';  
    $row['v_oid'] = date('Ymd').mt_rand(1000,9999);  
    $row['v_mid'] = '20272562';  
    $row['v_url'] = 'http://ddd.com/touzi/' . $req->pid;  
    $row['key'] = '%()#QOKFDLS:1*&U';  
  
    $row['v_md5info'] = strtoupper(md5(implode('', $row)));  
    return view('pay', $row);  
}
```

支付肯定有form表单 文当中已准备好

```
<form method=post action="http://127.0.0.1/myphp/pay/index.php">  
<p>商户编号<input type=text name=v_mid value="{{ $v_mid }}"></p>  
<p>订单编号<input type=text name=v_oid value="{{ $v_oid }}"></p>  
<p>订单总金额<input type=text name=v_amount value="{{ $v_amount }}"></p>  
<p>币种<input type=text name=v_moneytype value="{{ $v_moneytype }}"></p>  
<p><input type=hidden name=v_url value="{{ $v_url }}"><p>  
<p><input type=hidden name=v_md5info value="{{ $v_md5info }}"><p>  
    <input type="submit" name="name" value="支付">
```

http://127.0.0.1/myphp/pay/index.php模拟支付地址

防止站外token

有时候你可能希望设置一组并不需要 CSRF 保护的 URI。例如,如果你正在使用 Stripe 处理付款并使用了他们的 webhook 系统,你会需要从 CSRF 的保护中排除Stripe Webhook 处理程序路由,因为Stripe 并不会给你的路由发送 CSRF 令牌。你可以把这类路由放到 routes/web.php 外,因为 RouteServiceProvider 的 web 中间件适用于该文件中的所有路由。不过,你也可以通过将这类 URI 添加到 VerifyCsrfToken 中间件中的 \$except 属性来排除对这类路由的 CSRF 保护:

```
//app\Http\Middleware\VerifyCsrfToken.php  
protected $except = [  
    //  
    'touzi/*'  
];  
}
```

20.3整合到支付项目

上面的代码直接写入控制器中,判断我们支付的数据传到数据库中

```
$md5 =$req->v_oid . $req->v_pstatus . $req->v_amount . $req->v_moneytype .'()#QOKFDLS:1*&U';
$md5 = strtoupper(md5($md5));

if($md5 != $req->v_md5str){
    return "签名错误";
}

$user = $req->user();
$pro = Pro::find($pid);

$bid->pid = $pid;
$bid->uid = $user->uid;
$bid->title = $pro->title;
$bid->money = $req->v_amount*100;
$bid->pubtime = time();
.....
.....
```