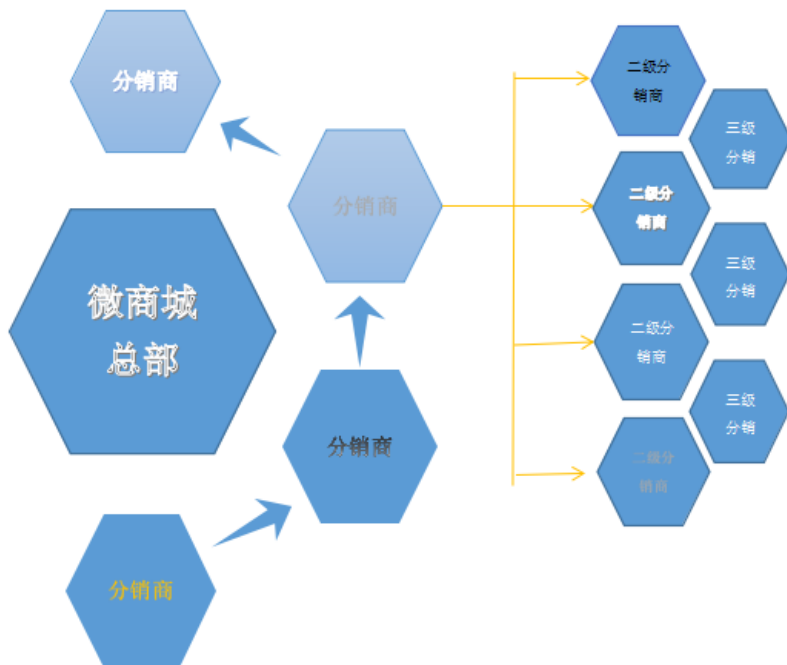


微分销商城

By 布尔教育 www.itbool.com

第1章 项目分析

1.1 何为分销



三级分佣,无限裂变,利用社交媒体的巨大流量做产品分销,分裂出成千上万个分销商,扩大销售规模,这是微信三级分销的核心价值之所在。微信三级分销平台,去掉了传统之间,为生产商带来分销商,使得生产商与消费者直接沟通,不仅降低生产商交易成本,而且为商家带来更多精准用户。同时,三级分销还为消费者提供成为分销商的机会,为消费者带来额外收入,间接提高消费者与生产商之间的粘度。

1.2 关键技术

- 场景二维码
- 用户上下级关系
- 微信登录

第2章 部署项目

2.1 安装laravel

```
composer create-project laravel/laravel=5.5 fenxiao
```

2.2 安装Wechat扩展

```
composer require overtrue/wechat
```

2.3 EasyWechat扩展预览

EasyWechat扩展 <https://easywechat.org/>

系列教程 <https://github.com/overtrue/wechat/wiki>

首先阅读开发文档 <https://easywechat.org/zh-cn/docs/>

如下图:

▼ 开始使用

概述

安装

在框架中使用

常见问题汇总

- ▶ 公众号
- ▶ 微信支付
- ▶ 小程序
- ▶ 开放平台
- ▶ 企业微信
- ▶ 自定义
- ▶ 其他

EasyWeChat 是一个开源的 微信 非官方 SDK。

EasyWeChat 的安装非常简单，因为它是一个标准的 Composer 包，这意味着任何满足下列安装条件的 PHP 项目支持 Composer 都可以使用它。

环境需求

- PHP >= 7.0
- PHP cURL 扩展
- PHP OpenSSL 扩展
- PHP SimpleXML 扩展
- PHP fileinfo 拓展

加入我们

 加入QQ群 ID: 319502940

第3章 接入微信

3.1 配置url

- 配置虚拟主机

```
<VirtualHost *:80>
    DocumentRoot "/xampp/htdocs/myphp/fenxiao/public"
    ServerName    bool.ittun.com
</VirtualHost>
```

- 添加路由

```
//准备控制器 php artisan make:controller WxController
//Routes/web.php
Route::any('/wx', 'WxController@index');
```

- 服务端验证
<https://www.easywechat.com/docs/master/zh-CN/official-account/tutorial>
- 写控制器

```
use EasyWeChat\Factory;

$config = [
    'app_id' => 'wx3cf0f39249eb0xxx',
    'secret' => 'f1c242f4f28f735d4687abb469072xxx',

    'response_type' => 'array', // 返回的数据类型

    'log' => [
        'level' => 'debug',
        'file' => __DIR__.'/wechat.log',
    ],
];

$app = Factory::officialAccount($config);

$response = $app->server->serve();
```

```
// 将响应输出
return $response;
```

- 测试号接口调试

因为我们写的服务端验证是在/wx路由下

接口配置信息修改

请填写接口配置信息，此信息需要你有自己的服务器资源，填写的URL需要正确响应微信发送的Token验证，请阅读[消息接口使用指南](#)。

URL http://bool.ittun.com/wx

Token buerjiaoyu

第4章 关注微信号

4.1 关注回复

-阅读文档 事件类型

-在 SDK 中使用 `$app->server->push(callable $callback)` 来设置消息处理器:

```
//在服务端监听事件,消息
$app = Factory::officialAccount($config);
$this->app->server->push(function ($message) {
//注意这里的$message是用户发的消息,不仅仅是文本,有可能是事件
//判断用户发过来的消息类型
    if($message['MsgType'] == 'event'){
//判断用户发过来的是什么事件类型
        if($message['Event'] == 'subscribe'){
//返回给用户消息,但是注意是xml类型
            return "欢迎关注布尔教育!!!";
        }
    }
});
//服务器端处理xml数据,并返回给用户
$response = $app->server->serve();
return $response;
```

写好代码仍然收不到回复,我们可以查看日志

laravel: /storage/logs/laravel.log

发现, POST提交的csrf_token参数缺失, 配置wx不验证token即可

/app/Http/Middleware/VerifyCsrfToken.php

```
protected $except = [
    //
    '/wx',
];
```

然后正常取关重关即可

4.2 把功能移至方法中

因为在框架中开发,保持代码的可观性

```
//在服务端监听事件,消息
//在服务端监听事件,消息
$app = Factory::officialAccount($config);
$this->app->server->push(function ($message) {
//注意这里的$message是用户发的消息,不仅仅是文本,有可能是事件
//判断用户发过来的消息类型
    if($message['MsgType'] == 'event'){
```

```

        return $this->shijian($message);
    }else{
        return $this->xiaoxi($message);
    }

});
//服务器端处理xml数据,并返回给用户
$response = $app->server->serve();
return $response;

```

```

public function shijian($message){
    //判断用户发过来的是什么事件类型
    if($message['Event'] == 'subscribe'){
        //返回给用户消息,但是注意是xml类型
        return "欢迎关注布尔教育!!!";
    }
}

public function xiaoxi($message){
    if($message['MsgType'] == 'text'){
        return '你好!!!';
    }
}

```

4.3 创建user表和user Model

配置数据库

修改自带的user迁移文件

```

public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->increments('uid');
        $table->string('name');
        $table->string('openid');
        $table->integer('subtime');
    });
}

```

//执行迁移命令 `php artisan migrate`

删除原有的user Model并自行创建 Model并修改

App\User.php

```

namespace App;

use Illuminate\Database\Eloquent\Model;

class User extends Model
{
    //
    protected $primaryKey = 'uid';
    public $timestamps = false;
}

```

注意数据库的版本

4.4 获取用户详细信息

- 阅读文档 获取用户的信息

获取用户基本信息(UnionID机制)

在关注者与公众号产生消息交互后，公众号可获得关注者的OpenID（加密后的微信号，每个用户对每个公众号的OpenID是唯一的。对于不同公众号，同一用户的openid不同）。公众号可通过本接口来根据OpenID获取用户基本信息，包括昵称、头像、性别、所在城市、语言和关注时间。

请注意，如果开发者有在多个公众号，或在公众号、移动应用之间统一用户帐号的需求，需要前往微信开放平台（open.weixin.qq.com）绑定公众号后，才可利用UnionID机制来满足上述需求。

用户信息的获取是微信开发中比较常用的一个功能了，以下所有的用户信息的获取与更新，都是基于微信的 `openid` 的，并且是已关注当前账号的，其它情况可能无法正常使用。

获取用户信息

获取单个：

```
$user = $app->user->get($openid);
```

获取多个：

```
$users = $app->user->select([$openid1, $openid2, ...]);
```

```
public $app = null;
//在__construct中实例化Application,并赋给$app属性,便于在所有方法中引用
public function __construct(){
//....
//在__construct中实例化Application,并赋给$app属性,便于在所有方法中引用
    $this->app = Factory::officialAccount($config);
}
public function shijian($message){
    //判断用户发过来的是什么事件类型
    if($message['Event'] == 'subscribe'){
        //获取用户信息,根据用户的openid获取,昵称,openid,等
        $openid = $message['FromUserName'];
        //获取关注用户详细信息 Userservice实例
        $userservice = $this->app->user;
        $userinfo = $userservice->get($openid);
        //获取用户昵称
        $name = $userinfo['nickname'];
        //取关重关,成功返回,则调用UserService实例成功
        return $name;
    }
}
```

- 注意

调用easywechat用的是curl去发送post的请求,可能会有以下的错误

```
curl: (60) SSL certificate problem: unable to get local issuer certificate
```

这是因为缺少CA证书

解决办法

1. 下载 CA 证书

你可以从 <http://curl.haxx.se/ca/cacert.pem> 下载 或者 使用[微信官方提供的证书](#)中的 CA 证书 `rootca.pem` 也是同样的效果。

2. 在 `php.ini` 中配置 CA 证书

只需要将上面下载好的 CA 证书放置到您的服务器上某个位置，然后修改 `php.ini` 的 `curl.cainfo` 为该路径（**绝对路径！**），重启 `php-fpm` 服务即可。

```
curl.cainfo = /path/to/downloaded/cacert.pem
```

注意证书文件**路径为绝对路径！**以自己实际情况为准。

4.5 关注用户写入数据库

```
public function shijian($message){
    //判断用户发过来的是什么事件类型
    if($message['Event'] == 'subscribe'){
        //获取用户信息,根据用户的openid获取,昵称,openid,等
        //....
        //写入数据库
        $userModel = new User();
        $userModel->name = $name;
        $userModel->openid = $openid;
        $userModel->subtime = time();
        $userModel->save();
        //返回给用户消息,但是注意是xml类型
        return "欢迎关注布尔教育!!!";
    }
}
```

4.6 取消关注

建立添加列迁移文件

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->boolean('status')->default(1);
    });
}
```

```
/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('status');
    });
}
```

```
});  
}
```

关注用户判断是事件类型是subscribe,那么取关的时候,需要判断事件类型是unsubscribe

```
public function shijian($message){  
    $userModel = new User();  
    //获取用户信息,根据用户的openId获取,昵称,openid,等  
    $openid = $message['FromUserName'];  
    //判断用户发过来的是什么事件类型  
    if($message['Event' == 'subscribe'){  
  
        //.....  
        //.....  
  
        //如果重新关注,查询有没有此用户用户的状态,查询,修改  
        $Quser = $userModel->where('openid',$openid)->first();  
        if($Quser){  
            $Quser->status = 1;  
            $Quser->save();  
        }else{  
            //写入数据库  
            $userModel->name = $name;  
            $userModel->openid = $openid;  
            $userModel->subtime = time();  
            $userModel->save();  
        }  
        //返回给用户消息,但是注意是xml类型  
        return "欢迎关注布尔教育!!!";  
    }elseif($message['Event'] == 'unsubscribe'){  
        //如果取关,修改用户的状态,查询,修改  
        $Quser = $userModel->where('openid',$openid)->first();  
        if($Quser){  
            $Quser->status = 0;  
            $Quser->save();  
        }  
    }  
}
```

4.7 生成场景二维码

- 手册->二维码

```
public function qr($openid){  
    //获取创建二维码实例  
    $qrcode = $this->app->qrcode;  
    //创建永久二维码  
    $result = $qrcode->forever($openid);  
    $ticket = $result['ticket'];  
    //获取二维码内容  
    $url = $qrcode->url($ticket);  
    $content = file_get_contents($url); // 得到二进制图片内容  
    $dir = $this->mkd() . '/' . $openid . '.jpg';  
    file_put_contents(public_path() . $dir , $content); // 写入文件  
}  
//生成层次的文件夹  
public function mkd(){  
    $today = date('/Y/m/d');  
    if(!is_dir( public_path() . $today )){  
        mkdir(public_path() . $today , 0777 ,true);  
    }  
    return $today;  
}  
}
```

第5章 三级分销关系

5.1 表结构

用户自己的上级,上级的上级,如何去设计表结构
无限极分类?如下图

uid	name	pid
3	lisi	2
4	wang	3
5	wang	4

有无限级分类没有必要:

- 1. 查询麻烦,只能查看自己的上级.
- 2. 查上上级,上上上级需要循环或递归.
- 3. 事实上不用无限级,3级就足够.

因此,我们直接把会员的上级,上上级,上上上通过3个字段,记录完毕.如下:

uid	name	p1	p2	p3
1	tom	0	0	0
2	poly	1	0	0
3	lisi	2	1	0
4	wang	3	2	1
5	lucy	4	3	2

5.2 迁移文件的准备

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->integer('p1')->default(0);
        $table->integer('p2')->default(0);
        $table->integer('p3')->default(0);
    });
}
```



```

    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('p1');
        $table->dropColumn('p2');
        $table->dropColumn('p3');
    });
}

```

5.3 用户上下级关系生成

```

public function shijian($message){
    $openid = $message['FromUserName'];
    $user = new User;
    if($message['Event'] == 'subscribe'){
        //判断当前数据库中是否有用户的数据
        $Quser = User::where('openid',$openid)->first();
        if($Quser){
            $Quser->status = 1;
            $Quser->save();
        }else{
            //获取用户信息
            $userinfo = $this->app->user->get($openid);
            $user->name = $userinfo['nickname'];
            $user->openid = $openid;
            $user->pubtime = time();
            //扫描带参数的二维码
            if($message['EventKey']){
                $openstr = str_replace('qrscene_', '', $message['EventKey']);
                $openUser = User::where('openid',$openstr)->first();
                $user->p1 = $openUser->uid;
                $user->p2 = $openUser->p1;
                $user->p3 = $openUser->p2;
            }
            $user->save();
        }
        $this->qrcode($message);
        return '欢迎关注!!!';
    }else{
        $this->unsub($message);
    }
}

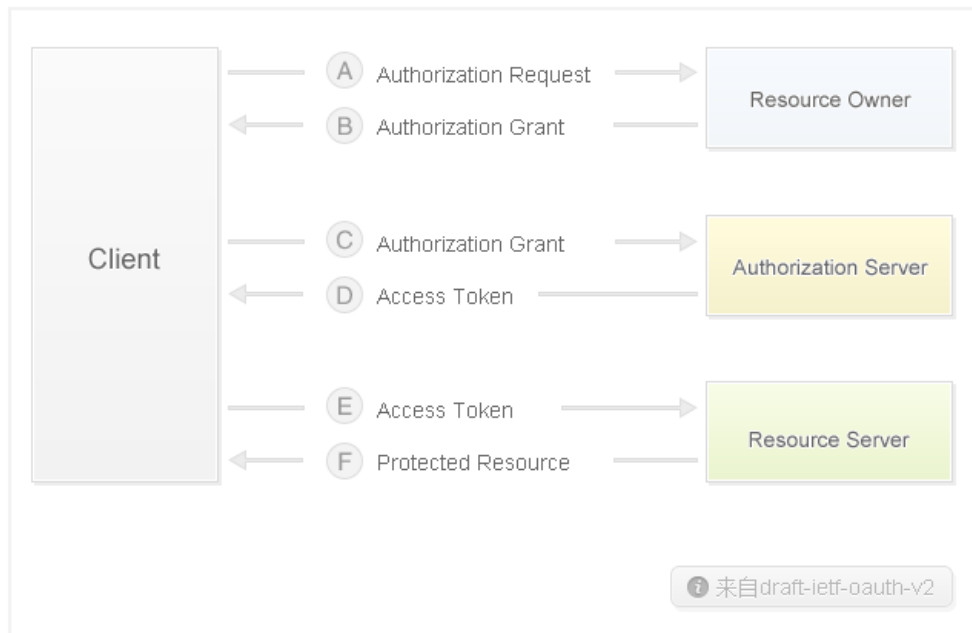
```

上下级关系

第6章 第三方授权登录

6.1 OAuth2.0协议

OAuth2.0协议授权官方流程图



关键点解释:

- 用户打开网站点击微博登录,网站携带授权及回调地址等信息发送请求到微博;
- 微博弹出登录界面,用户登录微博,微博发送数据到指定回调地址到网站;
- 网站接收数据,使用CURL的携带刚返回的数据再次请求微博,微博返回网站身份确认信息;
- 网站携带确认信息,再次请求微博,微博返回授权用户信息

6.2 申请微博授权

- 申请微博授权,填写回调地址,拿到 App Key 和App Secret

网站控制台

数据统计

部署微连接

网站信息

接口管理

已有权限

申请权限

调用频次

授权机制

管理网站

OAuth2.0 授权有效期

测试

普通

当前级别: 测试

应用未通过文案来源审核,不可以申请更高级别的授权有效期

查看应用状态

对应授权有效期: 1天

授权级别	测试	普通
授权有效期	1天	30天

OAuth2.0 授权设置

授权回调页: <http://bool.ittun.com/wx>

取消授权回调页: <http://bool.ittun.com>



微博登录



微博登录包括身份认证、用户关系以及内容传播。允许用户使用微博帐号登录访问第三方网站，分享内容，同步信息。它有如下特点：

- 直接用微博帐号登录你的网站
- 获得高质量微博活跃用户
- 更便捷的分享到微博，提高网站信息传播力

了解更多：[微博登录详细介绍](#) [微博openAPI文档](#) [相关案例](#)

如果您已经了解如何使用微博OpenAPI开发，那么您一定需要下面的信息：

当前域名：`www.jiyongkun.top`

App Key:
`1787884858`

App Secret:
`111c58e06a63fe2bb2bb5c68f7a5083d`

以上信息是基于微博openAPI进行开发，把您的网站和微博连接的重要信息，请注意不要随意泄露

- 查看文档:微博登录
接口文档

接口	说明
<code>OAuth2/authorize</code>	请求用户授权Token
<code>OAuth2/access_token</code>	获取授权过的Access Token
<code>OAuth2/get_token_info</code> 新	授权信息查询接口
<code>OAuth2/revokeoauth2</code> 新	授权回收接口
<code>OAuth2/get_oauth2_token</code>	OAuth1.0的Access Token更换至OAuth2.0的Access Token

注意事项

1. OAuth2.0授权无需申请，任何应用都可以使用。如果开发者需要更长的授权有效期参考本文档授权有效期部分。
2. 如果你是站外网页应用或客户端应用，出于安全性考虑，需要在平台网站填写redirect_url（授权回调页），才能使用OAuth2.0，填写地址：“[我的应用](#)>[应用信息](#)>[高级信息](#)”，对于客户端，我们也提供了默认的回调页地址。详细请查看授权页功能部分

6.3 实现微博登录

6.3.1 请求用户授权token

- 阅读文档
Url 首先去携带参数,去请求Url
`https://api.weibo.com/oauth2/authorize`
请求方式
`Get/Post`

请求参数

	必选	类型及范围	说明
client_id	true	string	申请应用时分配的AppKey。
redirect_uri	true	string	授权回调地址，站外应用需与设置的回调地址一致，站内应用需填写 canvas page 的地址。
scope	false	string	申请scope权限所需参数，可一次申请多个scope权限，用逗号分隔。 使用文档
state	false	string	用于保持请求和回调的状态，在回调时，会在Query Parameter中回传该参数。开发者可以用这个参数验证请求有效性，也可以记录用户请求授权页前的位置。这个参数可用于防止跨站请求伪造（CSRF）攻击
display	false	string	授权页面的终端类型，取值见下面的说明。
forcelogin	false	boolean	是否强制用户重新登录，true：是，false：否。默认false。
language	false	string	授权页语言，缺省为中文简体版，en为英文版。英文版测试中，开发者任何意见可反馈至 @微博API

返回数据

返回值字段	字段类型	字段说明
code	string	用于第二步调用oauth2/access_token接口，获取授权后的access token。
state	string	如果传递参数，会回传该参数。

- 实现请求url

准备回调url路由 `Route::any('weibo', 'WxController@weibo');`

准备路由 `Route::get('denglu', 'WxController@denglu');`

准备模板 `login.blade.php`

```
<body>
<h1>实现微博登录</h1>
<a href="https://api.weibo.com/oauth2/authorize?client_id=
1787884858&redirect_uri=http://bool.ittun.com/weibo">微博登录</a>
</body>
</html>
```

点击微博登录,登录成功之后会携带code参数跳转到回调url



那么在回调url中的weibo方法中获取code

```
public function weibo(){
    $weibo_code = $GET['code'];
    var_dump($weibo_code);
}
```

6.3.2 获取授权过的Access Token

- 阅读文档

用post的请求方式去携带参数请求url,获取Access Token

`https://api.weibo.com/oauth2/access_token`

请求参数

	必选	类型及范围	说明
client_id	true	string	申请应用时分配的AppKey。
client_secret	true	string	申请应用时分配的AppSecret。
grant_type	true	string	请求的类型，填写authorization_code

grant_type为authorization_code时

	必选	类型及范围	说明
code	true	string	调用authorize获得的code值。
redirect_uri	true	string	回调地址，需需与注册应用里的回调地址一致。

返回数据

```
{
  "access_token": "ACCESS_TOKEN",
  "expires_in": 1234,
  "remind_in": "798114",
  "uid": "12341234"
}
```

```
$weibo_code = $_GET['code'];
//使用Curl请求url
$url = 'https://api.weibo.com/oauth2/access_token';
$data = [
  'client_id' => '1787884858',
  'client_secret'=>'111c58e06a63fe2bb2bb5c68f7a5083d',
  'grant_type'=>'authorization_code',
  'code'=>$weibo_code,
  'redirect_uri'=>'http://bool.ittun.com/weibo'
];
$ch = curl_init($url);
curl_setopt($ch, CURLOPT_POST, 1);
//post变量.一定要用http_build_query处理下请求
curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
curl_setopt($ch,CURLOPT_RETURNTRANSFER ,1);
$output = curl_exec($ch);
curl_close($ch);
//打印获取到的token,返回的数据是json格式,并转化成数组
print_r(json_decode($output,true)['access_token']);
```

6.3.3获取用户授权信息

- 首先阅读文档,微博Api->用户接口->文档信息

微博API

粉丝服务接口	微博接口	评论接口	用户接口	关系接口
账号接口	收藏接口	搜索接口	提醒接口	短链接口
公共服务接口	位置服务接口	地理信息接口	地图引擎接口	支付接口
OAuth 2.0授权接口				

用户		
读取接口	users/show	获取用户信息
	users/domain_show	通过个性域名获取用户信息
	users/counts	批量获取用户的粉丝数、关注数、微博数

users/show

根据用户ID获取用户信息

URL

<https://api.weibo.com/2/users/show.json>

支持格式

JSON

HTTP请求方式

GET

请求参数

	必选	类型及范围	说明
access_token	true	string	采用OAuth授权方式为必填参数，OAuth授权后获得。
uid	false	int64	需要查询的用户ID。
screen_name	false	string	需要查询的用户昵称。

通过阅读文档,知道需要用get的请求方式请求url,
根据uid获取用户信息,所以只需要携带参数uid就可以

```
$rs = json_decode($output,true);
$token = $rs['access_token'];
$uid = $rs['uid'];

//凭借用户uid和access_token获取用户信息
$userinfo =file_get_contents('https://api.weibo.com/2/users/show.json?
access_token='.$token.'&uid='.$uid);
echo $userinfo;
```

返回字段说明

返回字段	字段类型	字段说明
id	int64	用户UID
idstr	string	字符串型的用户UID
screen_name	string	用户昵称
name	string	友好显示名称
province	int	用户所在省级ID
city	int	用户所在城市ID
location	string	用户所在地
description	string	用户个人描述
uri	string	用户博客地址
profile_image_url	string	用户头像地址（中图），50×50像素
profile_url	string	用户的微博统一URL地址
domain	string	用户的个性化域名
weihao	string	用户的微号
gender	string	性别，m：男、f：女、n：未知
followers_count	int	粉丝数
friends_count	int	关注数
statuses_count	int	微博数
favourites_count	int	收藏数
created_at	string	用户创建（注册）时间
following	boolean	暂未支持
allow_all_act_msg	boolean	是否允许所有人给我发私信，true：是，false：否
geo_enabled	boolean	是否允许标识用户的地理位置，true：是，false：否
verified	boolean	是否是微博认证用户，即加V用户，true：是，false：否
verified_type	int	暂未支持
remark	string	用户备注信息，只有在查询用户关系时才返回此字段
status	object	用户的最近一条微博信息字段 详细
allow_all_comment	boolean	是否允许所有人对我的微博进行评论，true：是，false：否
avatar_large	string	用户头像地址（大图），180×180像素
avatar_hd	string	用户头像地址（高清），高清图像原图
verified_reason	string	认证原因
follow_me	boolean	该用户是否关注当前登录用户，true：是，false：否
online_status	int	用户的在线状态，0：不在线、1：在线
bi_followers_count	int	用户的互粉数
lang	string	用户当前的语言版本，zh-cn：简体中文，zh-tw：繁体中文，en：英语

第7章 微信授权登录

在开始之前请一定要记住,先登录公众号后台,找到边栏“开发”模块下的“接口权限”,点击“网页授权获取用户基本信息”后面的修改，添加你的网页授权域名。

查看手册得知,我们需要以下步骤：

- 给微信添加授权后的回调域名
- 发送授权请求
- 授权验证(扩展已经实现,不用我们关心)
- 授权成功后，回调页面的授权检测
- 路由配置

```
Route::any('center','UserController@center');
Route::any('login','UserController@login');
Route::any('logout','UserController@logout');
```

- 控制器准备

```
php artisan make:controller UserController --plain
```

- 阅读文档 完成微信授权

```
public $app = null;
public function __construct(){
    $config = [
        'app_id' => 'wx0ac6bb2f916d6c30123',
        'secret' => 'af0c5bd90cfe92f2da3b5538c3f4d861235',

        'response_type' => 'array',

        'log' => [
            'level' => 'debug',
            'file' => base_path().'/wechat.log',
        ],
        'oauth' => [
```

```

        'scopes' => ['snsapi_userinfo'],
        'callback' => '/login',
    ],
];
$this->app = Factory::officialAccount($config);
}
//网页授权
public function center(Request $req){
    //判断如果没有用户登录跳转登录页面
    if(!$req->session()->has('wechat_user')){
        $oauth = $this->app->oauth;
        return $oauth->redirect();
    }
    return "欢迎登录!!!";
}
public function login(){
    $oauth = $this->app->oauth;
    $user = $oauth->user();//微信给我们提供用户信息
    session()->put('wechat_user',$user);//用户信息存入session
    //登录成功调转商城中心
    return redirect('center');
}

public function logout(){
    //用户退出清除session
    session()->forget('wechat_user');
}
}

```

第8章 商城建设

8.1 首页展示

- 路由准备

```

Route::get('/', 'GoodsController@index');
Route::get('insert', 'GoodsController@insert');

```

- 迁移文件准备
- `php artisan make:migration create_table_goods --create=goods`

```

public function up()
{
    Schema::create('goods', function (Blueprint $table) {
        $table->increments('gid');
        $table->string('goods_name');
        $table->string('goods_price');
        $table->string('goods_img');
    });
}

```

- 模板准备
修改css路径,图片路径等
- 方法准备

```

public function insert(){
    $data = [
        ['goods_name'=>'月季', 'goods_price'=>'23.8', 'goods_img'=>'goods_1.jpg'],
        ['goods_name'=>'玫瑰', 'goods_price'=>'45.6', 'goods_img'=>'goods_2.jpg'],
        ['goods_name'=>'桃花', 'goods_price'=>'30.8', 'goods_img'=>'goods_3.jpg'],
        ['goods_name'=>'妖姬', 'goods_price'=>'55.6', 'goods_img'=>'goods_4.jpg']
    ];
    DB::table('goods')->insert($data);
}
public function index(){

```



```
$goods = DB::table('goods')->get();  
return view('index',['goods'=>$goods]);  
}
```

- index模板修改

```
<div class="row">  
    @foreach($goods as $g)  
        <div class="col-xs-6 goods">  
            <a href="{url('goods',['gid'=>$g->gid])}"></a>  
            <p>  
                {{$g->goods_name}}  
                &yen;<span>{{$g->goods_price}}</span>  
            </p>  
        </div>  
    @endforeach  
</div>
```

- 本地测试
在本地测试如果ok,因为是在微信上操作,所以需要修改菜单接口,生成一个链接,点击直接跳转到商城首页,然后进行购买等
- 修改菜单接口



- 点击跳转

简洁大气的商城



月季 ¥23.8



玫瑰 ¥45.6



首页

个人中心

帮助

8.2 商品详情页

- 路由准备
- `Route::get('goods/{gid}', 'GoodsController@goods');`
- 方法准备

```
public function index(){
    $goods = DB::table('goods')->get();
    return view('index', ['goods'=>$goods]);
}
```

- 模板准备和修改

```
<div class="row">
    <div class="col-xs-12 goods">
        <a href="#"></a>
        <p>
            {{$goods_info->goods_name}}
            &yen;<span>{{$goods_info->goods_price}}</span>
        </p>
        <p>
            <a class="btn btn-primary" href="#">加入购物车</a>
        </p>
    </div>
</div>
```

第9章 购物流程

9.1 购物车类

- 安装购物车类

在<https://packagist.org> 搜索 cart 相关包

安装 `darryldecode/cart` .

```
composer require darryldecode/cart=~2.4
```

按说明配置:

在 `/app/config.php` 中的 Providers Array 中加一个单元

```
Darryldecode\Cart\CartServiceProvider::class
```

在 alias 别名数组中加一行

```
'Cart' => Darryldecode\Cart\Facades\CartFacade::class
```

9.2 添加及查看购物车

添加购物车

- 路由准备
- `Route::any('buy/{gid}', 'GoodsController@buy');`
- 方法准备

```
//添加购物车
public function buy($gid){
    $goods_info = DB::table('goods')->where('gid',$gid)->first();
    //使用Cart类的add方法添加
    Cart::add($gid, $goods_info->goods_name, $goods_info->goods_price, 1, array());
    //var_dump(Cart::getContent());exit;
    //加入成功,直接跳转购物车页面
    return redirect('cart');
}
```

查看购物车

- 路由准备
- `Route::get('cart', 'GoodsController@cart');`
- 方法准备

```
//查看购物车
public function cart(){
    //获取假如购物车内容
    $goods = Cart::getContent();
    //共计多少钱
    $total = Cart::getTotal();
    return view('cart' , ['goods'=>$goods , 'total'=>$total]);
}
```

- view

```
<table class="table">
    @foreach($goods as $g)
        <tr>
            <td>{{ $g->name }}</td>
            <td>{{ $g->price }}</td>
            <td>{{ $g->quantity }}</td>
        </tr>
    @endforeach
    <tr>
        <td colspan="3">小计:&yen;{{ $total }}元</td>
    </tr>
</table>
```

清空购物车

- 路由准备
- `Route::any('cart_clear','GoodsController@cart_clear');`
- 方法准备

```
//清空购物车
public function cart_clear(){
    Cart::clear();
    return redirect('cart');
}
```

9.3 订单表及迁移文件

- 订单表

字段	类型	说明
oid	primary key	主键
ordsn	string	订单号
uid	integer	用户uid
openid	string	用户Openid
xm	string	收货者姓名
address	string	收货地址
tel	string(11)	手机号
money	float(7,2)	订单金额
ispay	tinyint	是否支付
ordtime	integer	下单时间时间

- 商品表

字段	类型	说明
iid	primary key	主键
oid	integer	订单号
gid	integer	商品id
goods_name	string(40)	商品名称
price	float(7,2)	单价
amount	smallint	数量

- 生成迁移文件
- `php artisan make:migration create_table_orders --create=orders`
- `php artisan make:migration create_table_items --create=items`

```
//orders 订单表
Schema::create('orders', function (Blueprint $table) {
    $table->increments('oid');
    $table->string('ordsn');
    $table->integer('uid');
    $table->string('openid',32);
    $table->string('xm' , 15);
    $table->string('address' , 30);
    $table->string('tel' , 11);
    $table->float('money' , 7,2);
    $table->tinyinteger('ispay');
    $table->integer('ordtime')->unsigned();
});

//items 商品表
Schema::create('items', function (Blueprint $table) {
    $table->increments('iid');
    $table->integer('oid');
    $table->integer('gid');
    $table->string('goods_name',40);
    $table->float('price' , 7,2);
    $table->smallinteger('amount');
});
```

- 执行迁移文件
- `php artisan migrate`
- 创建model Order.php Item.php
- `php artisan make:model Order`
- `php artisan make:model Item`

```
//Order.php

class Order extends Model
{
    protected $primaryKey = 'oid';
    public $timestamps = false;
}

//Item.php

class Item extends Model
{
    protected $primaryKey = 'iid';
    public $timestamps = false;
}
```

9.4 订单入库

- 路由准备
- `Route::any('done','GoodsController@done')`
- 方法准备

```
//订单入库
public function done(Request $req){
    //获取用户登录信息 从session中取
    $u = session()->get('wechat_user');
    $openid = $u->getId();
    //获取数据库的用户信息
```

```

$userinfo = DB::table('users')->where('openid',$openid)->first();
$order = new Order();
//共计多少钱
$total = Cart::getTotal();
//写入数据库
$order->ordsn = date('YmdHis').mt_rand(10000,99999);
$order->uid = $userinfo->uid;
$order->openid = $userinfo->openid;
$order->xm = $req->xm;
$order->address = $req->address;
$order->tel = $req->mobile;
$order->money = $total;
$order->ispay = 0;
$order->ordtime = time();
$order->save();
//写入商品表信息
//获取假如购物车内容
$goods = Cart::getContent();
foreach($goods as $g){
    $item = new Item();
    $item->oid = $order->oid;
    $item->gid = $g->id;
    $item->goods_name = $g->name;
    $item->price = $g->price;
    $item->amount = $g->quantity;
    $item->save();
}
$this->cart_clear();
return view('zhifu',['oid'=>$order->oid]);
}

```

```

zhifu.blade.php
<form action="/pay" method="post">
{!!csrf_field()!!}
<input type="hidden" name="oid" value="{{ $oid }}">
<input class="btn btn-primary" type="submit" value="立即支付" >
</form>

```

- 支付路由准备
- `Route::any('pay', 'GoodsController@pay')`
- 方法准备

```

public function pay(Request $req){
    $oid = $req->oid;
    DB::table('orders')->where('oid',$oid)->update(['ispay'=>1]);
    return '购买成功';
}

```

第 10章 佣金分配

- fees表

字段	类型	说明
fid	integer	主键
oid	integer	订单号
byid	integer	买家id
uid	integer	受益者id
money	float	受益金额

- 迁移文件准备
- `php artisan make:migration create_table_fees --create=fees`

```
Schema::create('fees', function (Blueprint $table) {
    $table->increments('fid');
    $table->integer('oid');
    $table->integer('byid');
    $table->integer('uid');
    $table->float('money',7,2);
});
```

- model 准备
- `php artisan make:model Fee`
- 产生佣金

```
public function pay(Request $req){
    $oid = $req->oid;
    DB::table('orders')->where('oid',$oid)->update(['ispay'=>1]);
    //修改订单状态
    $order = Order::where('oid' , $req->oid)->first();
    $user = User::where('openid' , $order->openid)->first();
    //为上线生成收益
    //p1 0.4 p2 0.2 p3 0.1
    $shouyi = [$user->p1,$user->p2,$user->p3];
    $yj = [0.4,0.2,0.1];
    foreach ($shouyi as $k => $v) {
        if($v > 0){
            $fee = new Fee();
            $fee->oid = $order->oid; //订单ID
            $fee->byid = $user->uid; //买家ID
            $fee->uid = $v; //收益者
            $fee->money = $order->money * $yj[$k]; //收益额
            $fee->save();
        }
    }
    return '购买成功';
}
```