Please visit website: http://cxyroad.com

网站用户扫码公众号实现登录注册
==========

用户登录网站，想要实现扫一扫快捷登录注册，有两种方式可以完成。

1、注册开放平台，通过注册web应用，使用的SDK实现扫码登录注册。

2、注册公众号，利用公众号的消息事件通知机制来完成用户扫码并登录注册

本次网站开发使用第二种公众号扫码模式，公众号和开放平台应用各有不同，根据实际需求来申请并开发。

如何实现用户扫码公众号就能登录网站？ 这里首先把原理讲解一下，以便后面代码编写实施。

公众号有一种生成带参数的场景二维码功能，用户使用扫描场景二维码，会有一个扫码消息事件通知。我们在公众号后台基本配置中，有服务器回调地址可以配置URL，用来接收这种事件消息。然后根据收到的消息来完成网站用户的注册和登录逻辑。

简单来说，需要三步完成登录注册：

*1、服务端生成一个带参数的二维码，给前端并展示，前端要有二维码图和参数(ticket)。

*2、服务端收到事件后，将消息记录到数据库中(包含ticket)。

3、前端通过轮询的方式，定时请求服务端查询扫码结果(通过ticket)。

用户扫描带场景值二维码时，可能推送以下两种事件：

如果用户还未公众号，则用户可以公众号，后会将带场景值事件推送给开发者。

如果用户已经公众号，在用户扫描后会自动进入会话，也会将带场景值扫描事件推送给开发者。

生成带参数的二维码文档地址:

[developers.weixin.qq.com/doc/offiacc...](http://cxyroad.com/”https://developers.weixin.qq.com/doc/offiaccount/Account_Management/Generating_a_Parametric_QR_Code.html”)

以上原理了解后，下面就是具体实现的技术流程了，这里通过几步就可以完成。

第一步、登录公众号后台，配置基本信息。

![截图_20240704214424.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/75f58929a4234a6380d97e3f3022c468~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1920&h=1001&s=302001&e=jpg&b=fefefe)

![截图_20240704214534.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/d4a98d5e4ef4440f842d78bbc57ada70~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1919&h=996&s=247531&e=jpg&b=ffffff)

这里要注意以下，配置服务器回调URL，需要保证地址能正常访问且能正确响应。公众号系统会验证URL是否有效，还会验证是否正确解析消息和正确返回内容，来确定开发者具备开发能力。

具体文档地址:

[developers.weixin.qq.com/doc/offiacc...](http://cxyroad.com/”https://developers.weixin.qq.com/doc/offiaccount/Getting_Started/Getting_Started_Guide.html#_1-4-%E5%BC%80%E5%8F%91%E8%80%85%E5%9F%BA%E6%9C%AC%E9%85%8D%E7%BD%AE”)

官方代码示例用python， 我本次开发用PHP来实现(laravel 10)。

首先定义一个路由：

`Route::any('/wechat', 'WechatController@callback')->name('wechat.callback');`

接收消息并原样返回 echoStr

```
<?php

namespace App\Http\Controllers\Home;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Helpers\Log;

class WechatController extends Controller
{
    public function callback(Request $request)
{
        Log::info('WechatController', 'request -> ', $request->all());
        $signature = $request->input('signature');
        $timestamp = $request->input('timestamp');
        $nonce = $request->input('nonce');
        $echoStr = $request->input('echostr');
        if ($this->checkSignature($signature, $timestamp, $nonce)) {
            return $echoStr;
        } else {
            return 'success';
        }
    }

    /**
     * 官方提供的验签方法
     *
     * @param  string  $signature  签名
     * @param  string  $timestamp  时间戳
     * @param  string  $nonce  随机字符
     * @return  bool
     */
    private function checkSignature($signature, $timestamp, $nonce)
{
```

```
        $token = 'idcdcom';
        $tmpArr = array($token, $timestamp, $nonce);
        sort($tmpArr, SORT_STRING);
        $tmpStr = implode($tmpArr);
        $tmpStr = sha1($tmpStr);

        if ($tmpStr == $signature) {
            return true;
        } else {
            return false;
        }
    }
}
```

如果消息返回不正确，无法通过验证。

第二步、通过调用接口生成带参数的场景二维码

获取带参数的二维码的过程包括两步，首先创建二维码ticket，然后凭借ticket到指定URL换取二维码。

请求参数和响应结果示例参考便于理解

请求参数：

```
{
    "expire_seconds": "60",
    "action_name": "QR_STR_SCENE",
    "action_info": {
        "scene": {
            "scene_str": "login"
        }
    }
}
```

创建成功响应

```
{
    ”ticket”: ”gQGk7zwAAAAAAAAAAS5odHRwO****AAAA”,
    ”expire_seconds”: 60,
    ”url”: ”http://weixin.qq.com/q/02taMsVbKPb5e1yfF4hCcg”
}

```

字段解释：

![20240704215356.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/5a608c9e5f714ff6ace309b622502a4c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=917&h=846&s=265167&e=jpg&b=fefdfd)

业务实现代码：

```
<?php

namespace App\Services;

use Illuminate\Support\Facades\Cache;
use Illuminate\Support\Facades\Http;
use App\Models\UserWechat;
use App\Models\UserWechatScan;
use App\Helpers\Log;

class WechatService
{
    const FILENAME = ’WechatService’;
    const APPID = ’wx775c*****b1d50’;
    const SERECT = ’8ce00d3625*****4275239b002e61’;

    /**
     * 创建带参数的场景二维码
     *
     * @param  int  $seconds  二维码有效期
     * @return  object  二维码信息
     */
    static public function createQR($seconds = 60)
```

```
    {
        $url = 'https://api.weixin.qq.com/cgi-
bin/qrcode/create?access_token=' . self::getAccessToken();
        $params = [
            'expire_seconds' => $seconds,
            'action_name' => 'QR_STR_SCENE',
            'action_info' =>  ['scene' => [
                'scene_str' => 'login',
            ]],
        ];

        Log::info(self::FILENAME, 'createQR request -> ' .
json_encode($params));
        $response = Http::post($url, $params)->json();
        Log::info(self::FILENAME, 'createQR response -> ' .
json_encode($response));
        return $response;
    }

    /**
     * 处理通知事件
     *
     * @param  string  $postXML  通知消息xml字符串
     * @return  string  应答
     */
    static public function event($postXML)
    {
        //
{"ToUserName":"gh_880ace33e1ad","FromUserName":"xx","CreateTime
":"1719690890","MsgType":"event","Event":"SCAN","EventKey":"xxxx",
"Ticket":"xxdf"}
        $event = self::xmlToArray($postXML);

        // 事件
        if ($event['MsgType'] == 'event') {
            switch ($event['Event']) {
                case 'subscribe':
                    //
                    $user = [
                        'openid' => $event['FromUserName'],
                        'subscribe' => 1,
                        'created_at' => date('Y-m-d H:i:s',
$event['CreateTime']),
                    ];

                    if (UserWechat::where('openid', $event['FromUserName'])-
>exists()) {
                        UserWechat::where('openid', $event['FromUserName'])-
```

```
>update(['subscribe' => 1]);
                } else {
                    UserWechat::create($user);
                }

                // 如果是扫码
                if (isset($event['EventKey']) && isset($event['Ticket'])) {
                    echo self::handleScanQR($event);
                    return;
                }

                echo 'success';
                break;
            case 'unsubscribe':
                // 取消
                UserWechat::where('openid', $event['FromUserName'])-
>update(['subscribe' => 0]);
                echo 'success';
                break;
            case 'SCAN':
                // 扫码
                echo self::handleScanQR($event);
                break;
        }
    }

}

/**
 * 扫码成功 插入一条数据到数据库以便后面查询扫码结果
 *
 * @param  array  $event 解析XML后的数组
 * @return  string  应答消息字符串 消息回复到扫码人聊天窗口
 */
static public function handleScanQR($event)
{
    UserWechatScan::create([
        'openid' => $event['FromUserName'],
        'key' => $event['EventKey'],
        'ticket' => $event['Ticket'],
        'status' => 0,
    ]);

    return self::replyTextMsg($event['FromUserName'],
$event['ToUserName']);
}

/**
```

```
     * 回复文本消息
     *
     * @param  string  $toUser  发送消息接收对象
     * @param  string  $toUser  发送者
     * @return  string
     */
    static public function replyTextMsg($toUser, $fromUser)
    {
        // 回复文本消息
        $content = "您已扫码登录成功！\n登录时间: ". date('Y-m-d H:i:s') .
"\n使用中遇到任何问题，可以在公众号联系客服，您也可以添加我们的QQ和群。";
        $time = time();
        $xml = '<xml>
                <ToUserName><![CDATA['
    .$toUser.
    ']]></ToUserName>
                <FromUserName><![CDATA['
    .$fromUser.
    ']]></FromUserName>
                <CreateTime>'
    .$time.
    '</CreateTime>
                <MsgType><![CDATA[text]]></MsgType>
                <Content><![CDATA['
    .$content.
    ']]></Content>
                </xml>'
    ;
        return $xml;
    }

    /**
     * 获取 AccessToken
     *
     * @return  string  taoken字符串
     */
    static public function getAccessToken()
    {
        $cacheKey = 'wechat_access_token';

        if (Cache::has($cacheKey)) {
            return Cache::get($cacheKey);
        }

        $url = 'https://api.weixin.qq.com/cgi-bin/token';
        $params = [
            'grant_type' => 'client_credential',
```

```
            'appid' => self::APPID,
            'secret' => self::SERECT,
        ];

        Log::info(self::FILENAME, 'getAccessToken request -> ' .
json_encode($params));
        $response = Http::get($url, $params)->object();
        Log::info(self::FILENAME, 'getAccessToken response -> ' .
json_encode($response));
        if (isset($response->access_token)) {
            Cache::set($cacheKey, $response->access_token, 3600);
            return $response->access_token;
        }

        Log::error(self::FILENAME, 'getAccessToken error');
        return null;
    }

    /**
     * 获取用户个人信息
     *
     * @param  string  $openid  扫码用户的openid
     * @return  object  返回的用户信息
     */
    static public function getUserInfo($openid)
    {
        $url = 'https://api.weixin.qq.com/cgi-
bin/user/info?access_token=' . self::getAccessToken();
        $params = [
            'access_token' => self::getAccessToken(),
            'openid' => $openid,
            'lang' => 'zh_CN',
        ];

        Log::info(self::FILENAME, 'getUserInfo request -> ' .
json_encode($params));
        $response = Http::get($url, $params)->json();
        Log::info(self::FILENAME, 'getUserInfo response -> ' .
json_encode($response));
        return $response;
    }

    /**
     * simplexml_load_string 解析 XML 数据
     * @param string $postData 公众号响应的xml数据
     *
     * @return array
     */
```

```
    public static function xmlToArray($postXML)
    {
        try {
            //将xml进行解析为数组格式
            $data =
json_decode(json_encode(simplexml_load_string($postXML,
'SimpleXMLElement', LIBXML_NOCDATA)), true);
            Log::info(self::FILENAME, 'xmlToArray data -> ' .
json_encode($data));
            return $data;
        } catch (\Exception $e) {
            Log::info(self::FILENAME, 'xmlToArray error -> ' . $e-
>getMessage());
        }
    }

    /**
     * 官方提供的验签方法
     *
     * @param  string  $signature  签名
     * @param  string  $timestamp  时间戳
     * @param  string  $nonce  随机字符
     * @return  bool
     */
    private function checkSignature($signature, $timestamp, $nonce)
    {
        $token = 'idcdcom';
        $tmpArr = array($token, $timestamp, $nonce);
        sort($tmpArr, SORT_STRING);
        $tmpStr = implode($tmpArr);
        $tmpStr = sha1($tmpStr);

        if ($tmpStr == $signature) {
            return true;
        } else {
            return false;
        }
    }
}

```
```

这里额外提到一点，用户在扫码，收到事件消息通知，需要将消息数据库中，以便后需要前端查询结果使用。

数据表信息：
```

![截图_20240704215617.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/e70c9254b74a4d4d8acd5260c0e475c7~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1203&h=281&s=232485&e=jpg&b=fbf9f9)

openid： 扫码用户的唯一身份标识

key：场景的参数值

ticket：二维码的ticket,用来作为查询标识

status：查询到事件结果，标注下状态 0-初始化 1-查询完成

服务端还需要两个接口信息

1、生成参数场景二维码的接口

2、查询扫码结果的接口

```
// 公众号
Route::get('wechat/get-qrcode', 'WechatController@getQRcode')->name('get-qrcode'); // 生成带参数的二维码
Route::get('wechat/get-scan-result', 'WechatController@getScanResult')->name('get-scan-result'); // 查询扫码结果
```

这两个路由，控制器的代码就不贴了，生成QR上面的服务类有方法，查询也仅需要一条SQL。

至此， 服务端的工作已经完成。下面来看下前端界面作为参考吧。

第三步、前端展示二维码和轮询结果完成登录/注册

a、前端扫码登录界面：

![1.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/ea62f847e3d14366be715963fb08a03d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1920&h=1002&s=366063&e=jpg&b=d7d7d7)

弹出登录窗口的时候，前端发起一个请求 /get-qrcode 获取二维码信息，然后在有效期60秒内每2秒请求服务器一次，如果查询到事件，拿到返回的openid，完成登录操作。

b、如果60秒没有用户没有扫码，二维码过期

![2.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/7cb05e042f36419bafa9a28bb7ddafd8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1918&h=1002&s=558842&e=jpg&b=d0d0d0)

c、如果扫码成功完成登录

![3.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/2d64a91edebf40fb928a889a10ed49ee~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1920&h=999&s=575382&e=jpg&b=f8f8f8)

登录成功后就会收到公众号回复的消息了

原文链接: https://juejin.cn/post/7387542184472182820