## 微信支付

**一、WxPay.Config.php文件配置**

1、微信公众号：微信认证公众号，需开通微信支付的接口权限，是进行支付的场所。

公众号开发者中心->基本配置->AppId 和 AppSecret

const APPID = 'wx8397f8696b538317';//用来唯一标识公众号用户

const APPSECRET = '01c6d59a3f9024db6336662ac95c8e74'; //可以理解为密码

2、微信支付平台：付款后进行支付账单查询、提现退款等操作。

const MCHID = '1473426802'; //用来唯一标识微信商户用户

const MCHKEY = 'T6m9iK73b0kn9g5v426MKfHQH7X8rKwb'; //商户随时自行设置//仅JSAPI支付时需要配置

3、证书文件：存储私钥文件apiclient\_cert、存储公钥文件apiclient\_key

  商家在使用微信支付功能时进行身份验证的安全作用，但只在使用退款接口或者撤销订单时可能会用到（在接口函数中，可以选择不使用证书也能使用退款或撤销订单接口）。

证书下载:登录微信支付商户平台https://pay.weixin.qq.com/index.php/account/api\_cert，账户设置->API安全里面下载。

const SSLCERT\_PATH = '../cert/apiclient\_cert.pem'; //测试提供的私钥

const SSLKEY\_PATH = '../cert/apiclient\_key.pem'; //测试提供的公钥

注意:

（1）证书文件不能放在web服务器虚拟目录，应放在有访问权限控制的目录中，防止被他人下载；

（2）建议将证书文件名改为复杂且不容易猜测的文件名；

**二、公众号网页配置**

1、OpenId：微信用户id，用来唯一标识用户，即每次交易付款者；

在微信中有专门获取openid的接口，接口权限中设置可以获取用户ID信息权限的域名（每个用户对于不同公众号都会有一个特有ID，通过这个ID获取用户账号基本信息）

微信公众平台->开发->接口权限->网页授权获取用户基本信息，然后修改值为要获取OpenId的网页所在路径的域名(前面不用再添加http://或https://，也不要填域名ip)。

2、在微信支付按钮出设置微信支付授权目录（写到发起请求的控制器那一层）

  把微信支付的相关代码放到服务器里面,然后需要到公众平台授权该路径。

微信公众平台->微信支付功能->在开发配置中设置支付授权目录和测试目录，测试目录需在测试白名单中添加的测试账号才可以支付，而授权目录是所有人都可以支付的。

点击支付授权目录的修改,我们就可以设置授权目录了,注意发起支付请求的链接地址，都必须在支付授权目录之下,如把SDK全部放在WeChatPay 文件夹下,那么授权目录路径就是:www.xxxx.com:xx/WeChatPay，为了保证路径没错,拿SDK的文件夹example 为例,因为发起支付的文件 jsapi.php在example里面，可以多添加一个目录为:www.xxxx.com:xx/WeChatPay/example。

3、测试支付目录的修改参照授权目录，设置开发者微信账号为测试白名单（用微信开发者工具时需要）。

**三、代码修改**

1、SDK包

WxPay.Api.php"接口访问类，包含所有微信支付API列表的封装"

WxPay.Config.php"配置信息所在文件"

WxPay.Data.php"签名相关类，含签名生成"

WxPay.Exception.php"异常类"

WxPay.Notify.php"回调函数的父类"

文件夹-phpqrode存放了二维码功能相关文件

jsapi.php"网页调起支付页面"

native.php"付款二维码信息组装页面"

qrcode.php"生成二维码的页面"

native\_notify.php"扫描所生成二维码进入的页面"

notify.php"网页支付后的回调页面"

orderquery.php"订单查询页面"

download.php查退款单页面

refund.php 退款页面

refundquery.php 退款单查询的页面

WxPay.JsApiPay.php 微信浏览器打开网页支付核心类

WxPay.NativePay.php和WxPay.MicroPay.php 刷卡支付类

2、修改代码

（1）将 /lib/WxPay.Api.php 函数 postXmlCurl 中的两行代码

curl\_setopt($ch,CURLOPT\_SSL\_VERIFYPEER,TRUE);

curl\_setopt($ch,CURLOPT\_SSL\_VERIFYHOST,2);//严格校验

改为

curl\_setopt($ch,CURLOPT\_SSL\_VERIFYPEER,FALSE);

curl\_setopt($ch,CURLOPT\_SSL\_VERIFYHOST,FALSE);

为了禁止 cURL 验证对等证书（peer's certificate），否则二维码可能无法显示，输出结果会报错。

（2）(针对jsapi支付)如果访问jsapi.php时你要用get方式传递参数,那么你要去修改WxPay.JsapiPay.php中的

$baseUrl = urlencode('http://'.$\_SERVER['HTTP\_HOST'].$\_SERVER['PHP\_SELF'].$\_SERVER['QUERY\_STRING']);

改为:

$baseUrl = urlencode('http://'.$\_SERVER['HTTP\_HOST'].$\_SERVER['PHP\_SELF'].'?'.$\_SERVER['QUERY\_STRING']);

或者在写链接时写两个??传参,例如:http://www.test.com/demo/example/jsapi.php??id=xxx&..

3、处理支付结果

（1）在JavaScript处理支付结果

//调用微信JS api 支付，由于前端交互复杂，get\_brand\_wcpay\_request:cancel或者get\_brand\_wcpay\_request:fail可以统一处理为用户遇到错误或者主动放弃，不必细化区分。

function jsApiCall(){

WeixinJSBridge.invoke('getBrandWCPayRequest',<?php echo $jsApiParameters; ?>,function(res){

WeixinJSBridge.log(res.err\_msg);

alert(res.err\_code+res.err\_desc+res.err\_msg);

// res.err\_msg将在用户支付成功后返回ok，但并不保证它绝对可靠。

if(res.err\_msg == "get\_brand\_wcpay\_request：ok" ) {

// 支付成功

}else if((res.err\_msg =="get\_brand\_wcpay\_request：fail"){

// 支付失败

}else if((res.err\_msg =="get\_brand\_wcpay\_request：cancel"){

// 支付过程中用户取消

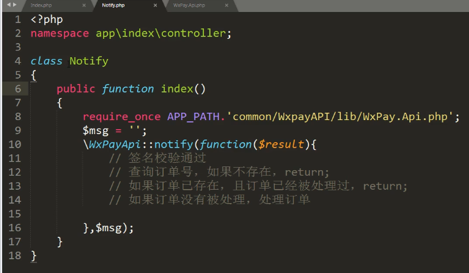
}

});

}

（2）处理异步通知

Index方法不能有授权验证，否则微信程序无法访问



$order = WxPayApi::orderQuery($input)); // 进行查询

var\_dump($order); // 打印出订单信息

   常用的订单信息:

if($order['err\_code\_des'] =="order not exist"){

// 订单不存在

}else{

$money = $order['total\_fee']; //所付款数,单位分

if($order['trade\_state'] =="SUCCESS"){

//支付成功

}else if($order['trade\_state'] =="REFUND"){

//已退款

}else if($order['trade\_state'] =="NOTPAY"){

//用户还没支付

}else if($order['trade\_state'] =="CLOSED"){

//订单关闭

}else if($order['trade\_state'] =="REVOKED"){

//已撤销（刷卡支付）

}else if($order['trade\_state'] =="USERPAYING"){

//用户支付中

}else if($order['trade\_state'] =="PAYERROR"){

//支付失败(其他原因，例如银行返回失败)

}

}

/\*\* 发起退款 \*/

$order = WxPayApi::refund($input);

/\*\* 在返回的数组中,我们能够获取键名return\_code \*/

if($order["return\_code"]=="SUCCESS"){

// 退款申请成功

}else if($order["return\_code"]=="FAIL"){

// 退款申请失败

}else{

// 未知状态

}

var\_dump($order);

}

确保已下订单

   在处理支付结果回调后，建议再进行一次订单查询，以确保万无一失，此外也可以设置一个订单恢复系统，如下：

if(isset($tradeId) && $tradeId != ""){

$input = new WxPayOrderQuery();

$input->SetOut\_trade\_no($tradeId); // 设置好要查询的订单

$order = WxPayApi::orderQuery($input)); // 进行查询

if($order['err\_code\_des'] =="order not exist"){

// 订单不存在

}else{

$money = $order['total\_fee']; //所付款数,单位分

if($order['trade\_state'] =="SUCCESS"){

//支付成功

//数据库操作

//两者对比

//若数据库中没记录，就恢复订单

}else if($order['trade\_state'] =="REFUND"){

//已退款

}else if($order['trade\_state'] =="NOTPAY"){

//用户还没支付

}else if($order['trade\_state'] =="CLOSED"){

//订单关闭

}else if($order['trade\_state'] =="REVOKED"){

//已撤销（刷卡支付）

}else if($order['trade\_state'] =="USERPAYING"){

//用户支付中

}else if($order['trade\_state'] =="PAYERROR"){

//支付失败(其他原因，例如银行返回失败)

}

}

}

4、支付后跳转

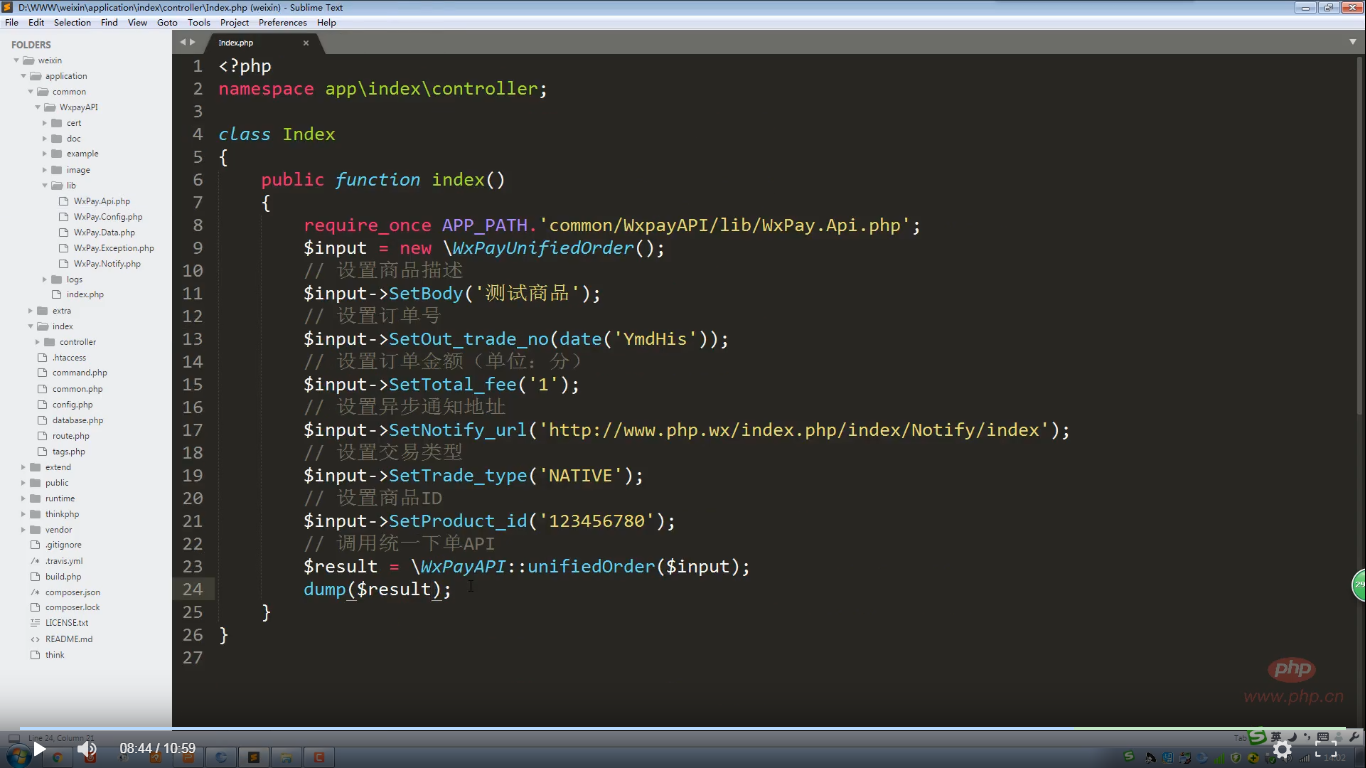
支付完成后,手机提示支付成功,并会显示支付信息,但是pc端的二维码页面却不会跳转.这需要在native.php页面设置一个定时器,设置为自己想要跳转到的页面路径即可:

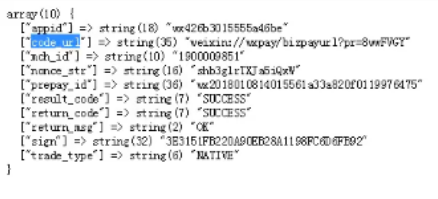


其中的$order\_id是get传过来的订单号.通过查询订单号对应商品的支付状态判断是否成功,然后跳转.

5、整合进TP的逻辑：  
　　前端微信支付按钮设置点击调用支付发起控制器方法、  
　　控制器运行，引用微信支付类、获取用户openid、获取订单数据、拼接出所有普通商户预支付jsp需要的数据，display出那个自定义的支付页面、在支付页面点击支付、调用微信提供的jspi发起支付的scripet函数发起支付、  
　　支付完成以后页面会重定向到（在自定义支付页面的script函数里设置的跳转目录{:U('controller/function)}）,并且异步（静默）设置的异步处理订单逻辑（如记录支付时间、标记为已经支付、标记是微信支付）之类的。

订单页面的微信支付按钮：  
 <a href="{:U('Wxpay/js\_api\_statr',array('order\_key\_num'=>$v['order\_key\_num']))}">微信支付</a>





## 支付同步与异步

一.概念

支付中的同步和异步：指用户支付后，服务器获取支付结果的两种不同方法。这两种方法可以单独使用，任意一种一般情况下都可以回调成功，为了安全和用户体验，两种回调方法并存。

同步：发送方发出数据后，等接收方发回响应以后才发下个数据包的通讯方式。 （打电话）

异步：发送方发出数据后，不等接收方发回响应，接着发送下个数据包的通讯方式。 （发短信）

二：区别（以支付宝支付为例）

1、同步回调只有一次且是最终结果，成功就是成功，失败就是失败；

异步回调可以有多次，异步回调如果失败，在失效时间内会继续回调，直到成功为止。

异步回调第一次收到订单信息是与返回页近乎同时，在判断不成功的情况下，会收到第二三次的通知，间隔从最先的一两分钟，到后面的几个小时。失效时间是48小时。

2、同步回调使用get请求传参；异步回调使用post请求传参。

3、同步是客户端回调；异步是服务器端回调。

## 比特币支付

1. 下载Coinbase的PHP库

https://github.com/coinbase/coinbase-php/releases

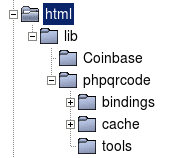
2. 下载PHP二维码库

http://phpqrcode.sourceforge.net/

3. 复制粘贴PHP库到你正开发APP的地方。

现在我们有了两个库，把它们的文件夹都移动到你想开发APP的那个文件夹。

文件夹的结构应该看起来像这样：

[](https://cdn.8btc.com/wp-content/uploads/2014/10/1-JU8WZw2D22kjXM8MWuoruA.png)

4. 确保你的PHP GD库是安装好了的。

5. 在你的Coinbase账户上创造一个新的API Key

在这部分你需要一个Coinbase账户，所以如果你没有的话，[点击这里注册](http://coinbase.com/)。

一旦你创建了一个账户，找到“设置”（ “Settings”） ，再找到“API Access”。

当创建新的API Key的时候，就在你初始的钱包里赋予这个key“地址”认可（ “addresses” permission） 。

6. 将我们的APP代码放进你的index.php文件里。

这段代码将你当前的地址从Coinbase上提取出来，然后检查你是否有余额。如果你没有余额，打赏按钮就会把赏金发到那儿。如果有余额的话，打赏按钮就会为你在Coinbase上创造一个新的地址，将赏金发给这个新地址。

<?php

$COINBASE\_API\_KEY = "YOUR-API-KEY";

$COINBASE\_API\_SECRET = "YOUR-API-SECRET";

require\_once("lib/Coinbase.php");

require\_once("lib/phpqrcode/qrlib.php");

$coinbase = Coinbase::withApiKey($COINBASE\_API\_KEY, $COINBASE\_API\_SECRET);

$current\_address = $coinbase->getReceiveAddress();

$currentbalance = $coinbase->getBalance($current\_address);

if ($currentbalance > 0) {

# get a new address

$tip\_address = $coinbase->generateReceiveAddress();

}else{

$tip\_address = $current\_address;

}

ob\_start();

QRcode::png("$tip\_address",null);

$qrcode = base64\_encode( ob\_get\_contents() );

ob\_end\_clean();

?>

<html>

<head>

<script type="text/javascript">

function display\_codes(){

var qrcode\_html = '<?php echo '<img src="data:image/png;base64,'.$qrcode.'\" width=150 height=150>';?>';

var address\_html = 'Thanks for your tip! You can send Bitcoins to the following address: <?php echo $tip\_address;?>';

document.getElementById('address').innerHTML = address\_html;

document.getElementById('qrcode').innerHTML = qrcode\_html;

}

</script>

</head>

<body>

<button onclick="display\_codes()">Tip me with Coinbase!</button>

<div id=”address”></div><br>

<div id=”qrcode”></div>

</body>

</html>