

分享-H5微信支付开发

1.申请微信支付帐户

商户在微信公众平台(申请扫码支付、公众号支付)或开放平台(申请APP支付)按照相应提示,申请相应微信支付模式。微信支付工作人员审核资料无误后开通相应的微信支付权限。微信支付申请审核通过后,商户在申请资料填写的邮箱中收取到由微信支付小助手发送的邮件,此邮件包含开发时需要使用的支付账户信息,见图3.1所示。

尊敬的【 **测试商户** 】：

恭喜您已成功通过微信支付商户资料审核！以下为您的账户所需要的重要信息，请注意保密：

微信支付商户号	
商户平台登录帐号	
商户平台登录密码	
申请微信支付公众号	
APPID	

[前往微信支付商户平台](#)

后续操作指引

前往公众平台完成入驻

您还需要登录微信公众平台，继续按操作指引完成微信支付入驻，请点击【公众平台-微信支付-支付申请-②签署承诺函】签署承诺函，并点击【公众平台-微信支付-支付申请-②签署协议】在线签署微信支付服务协议
[点击查看示意图](#)

根据开发文档进行开发

现在您已具备微信支付功能开发的资格，请您下载最新的文档进行开发

下载API证书、设置API密钥

开发所需的接口API证书、密钥，请您登录微信支付商户平台，在【账户设置-密码安全-API安全】中下载以及设置，如果你此前已经申请过微信支付，请您注意，新接口的API证书和密钥不会再随邮件下发

登录微信支付商户平台

商户平台是微信支付提供给您的商户管理软件系统，包括交易管理、结算管理、营销推广等。为了保证订单数据准确性以及退款的唯一性，请您只通过微信支付商户平台进行订单查询和退款操作

图3.1 微信审核通过邮件模板

邮件中的账户参数与接口API参数对应关系见表格3.1：

表3.1 账户参数说明

邮件中参数	参数名	详细说明
APPID	appid	appid是微信公众账号或开放平台APP的唯一标识，在公众平台申请公众账号或者在开放平台申请APP账号后，微信会自动分配对应的appid，用于标识该应用。可在微信公众平台-->开发者中心查看，商户的微信支付审核通过邮件中也会包含该字段值。
微信支付商户号	mch_id	商户申请微信支付后，由微信支付分配的商户收款账号。
密钥	key	交易过程生成签名的密钥，仅保留在商户系统和微信支付后台，不会在网络中传播。商户妥善保管该Key，切勿在网络中传输，不能在其他客户端中存储，保证key不会被泄漏。商户可根据邮件提示登录微信商户平台进行设置。也可按一下路径设置：微信商户平台 (pay.weixin.qq.com)->账户设置-->API安全-->密钥设置
Appsecret	secret	AppSecret是APPID对应的接口密码，用于获取接口调用凭证access_token时使用。在微信支付中，先通过OAuth2.0接口获取用户openid，此openid用于微信内网页支付模式下单接口使用。在开发模式中获取AppSecret（成为开发者且帐号没有异常状态）。

邮件中参数	参数名	详细说明
APPID	appid	， 用于标识该应用。可在微信公众平台
微信支付商户号	mch_id	， 用于标识该应用。可在微信公众平台
密钥	key	， 用于标识该应用。可在微信公众平台
Appsecret	secret	， 此

在微信公众号中申请微信支付的地方是：



2. 微信开发环境的配置

(1) 首先到微信公众号平台上配置一下，支付授权目录，如下图



(2) 然后到<https://pay.weixin.qq.com/index.php>，微信支付商户平台上配置一下api支付密钥

设置商户号支付密钥方法如下：

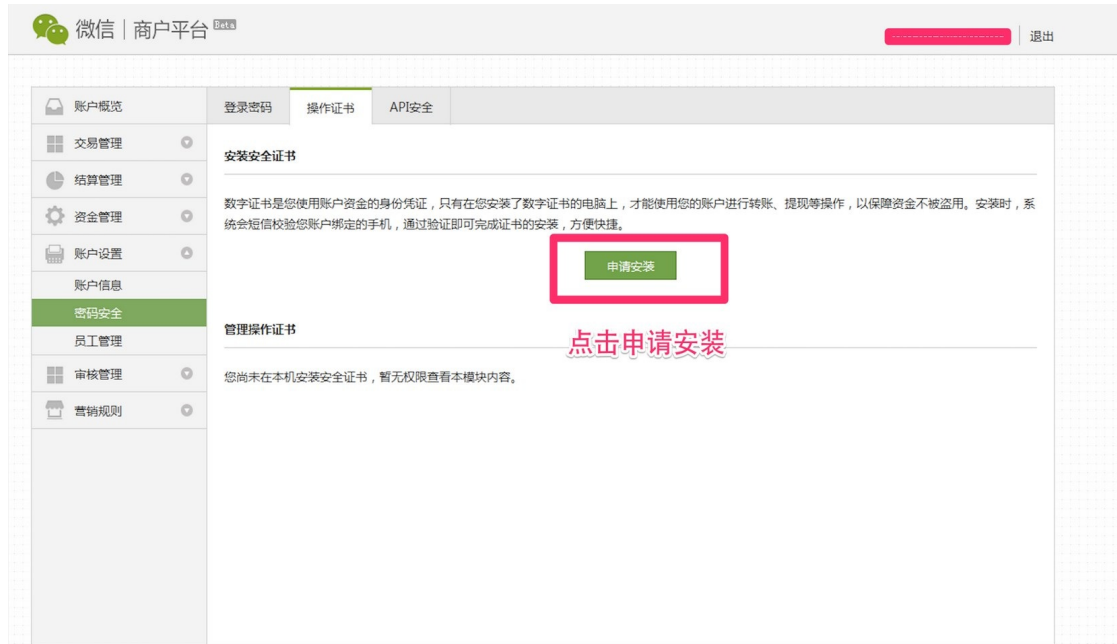
1. 申请通过审核后，打开微信发来的邮件。

直接获取微信支付商户号（接口文档中的商户号MCHID）；（也叫 受理商ID 填写到后台）。

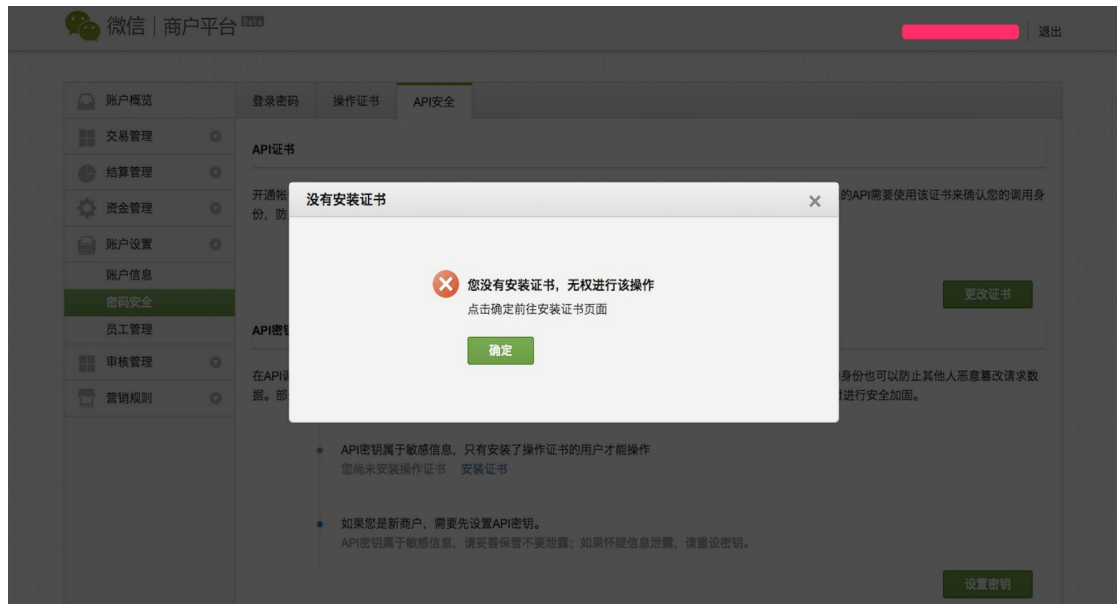
商户支付密钥key 按以下步骤自己设置生成。

2 点击访问 新商户平台<http://mch.weixin.qq.com> 或<https://pay.weixin.qq.com>，用账号和密码登录。并安装证书。

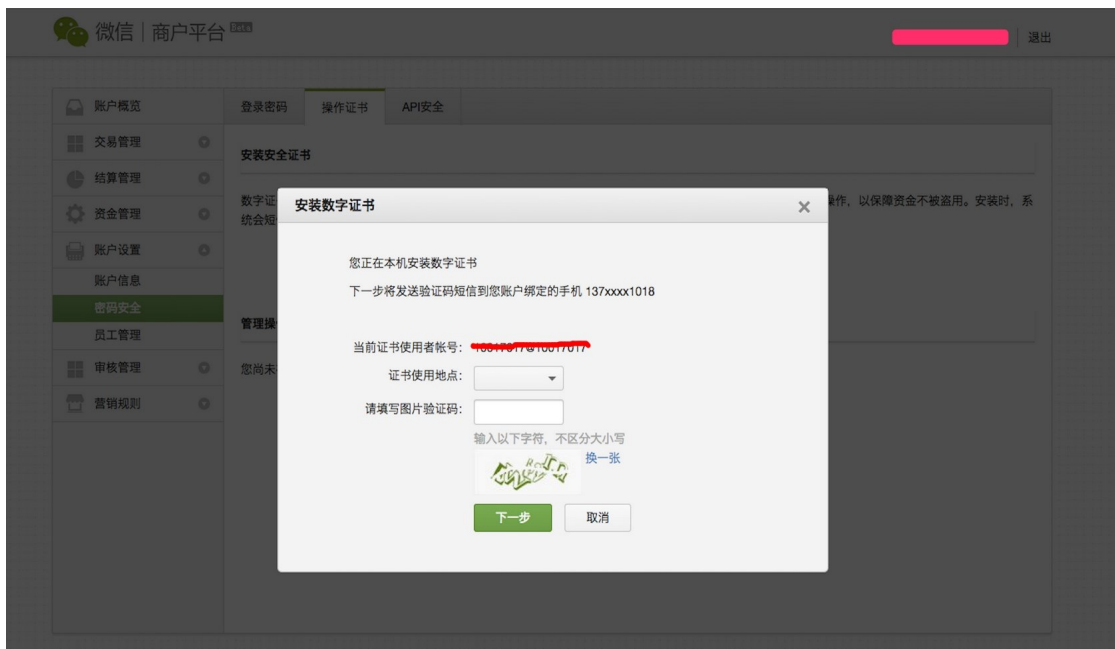
step 1



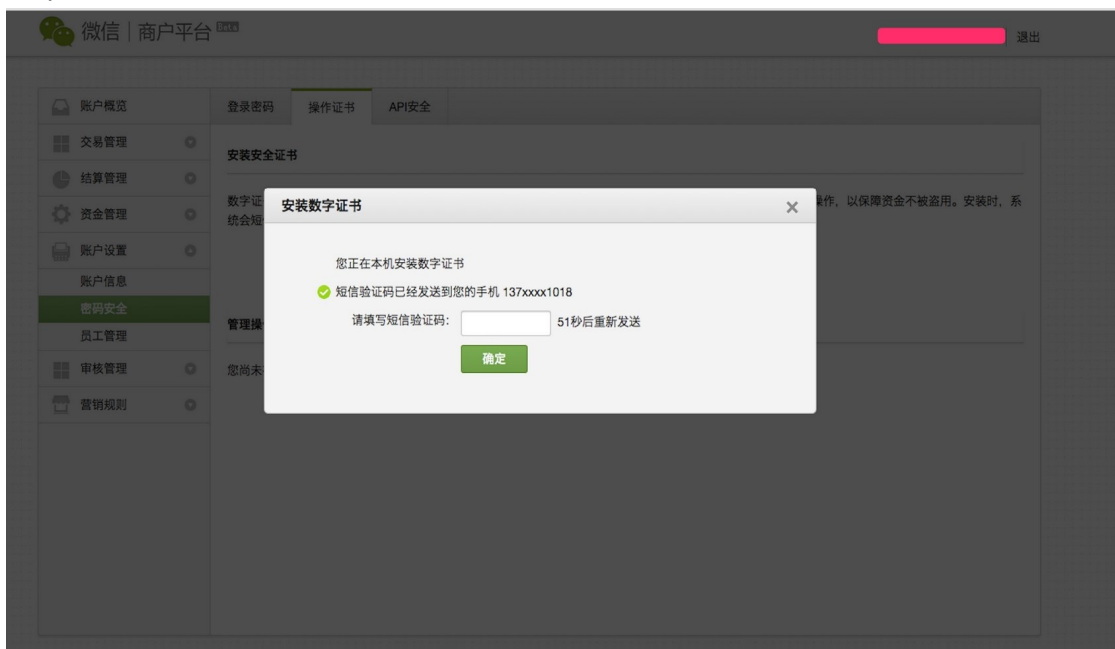
step 2 点击确定。



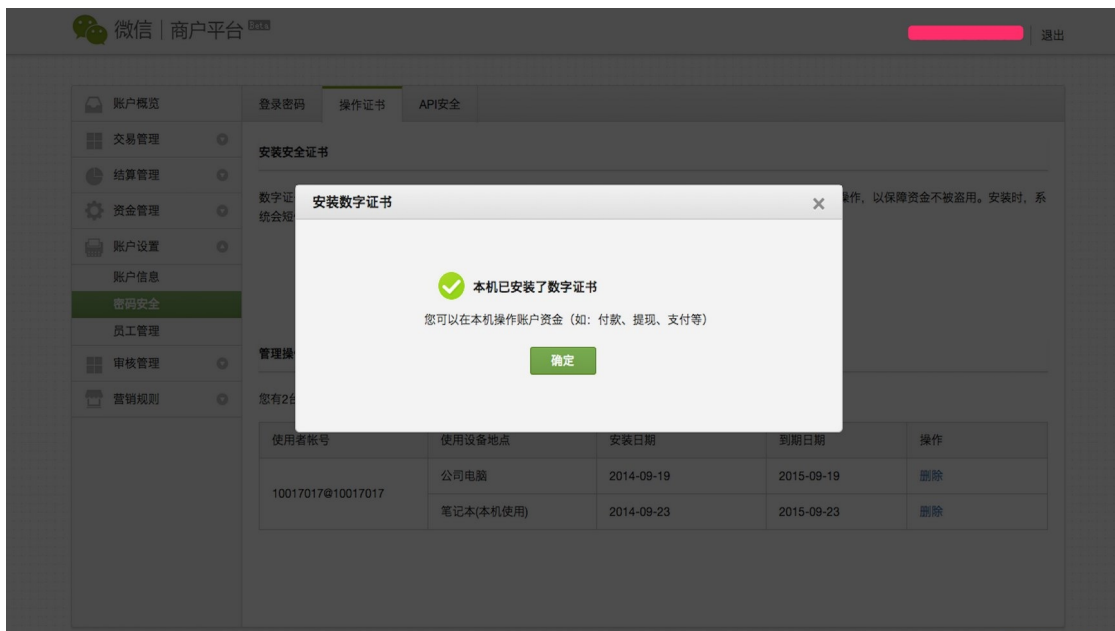
step 3 填写验证信息。



step 4 填写手机验证码。

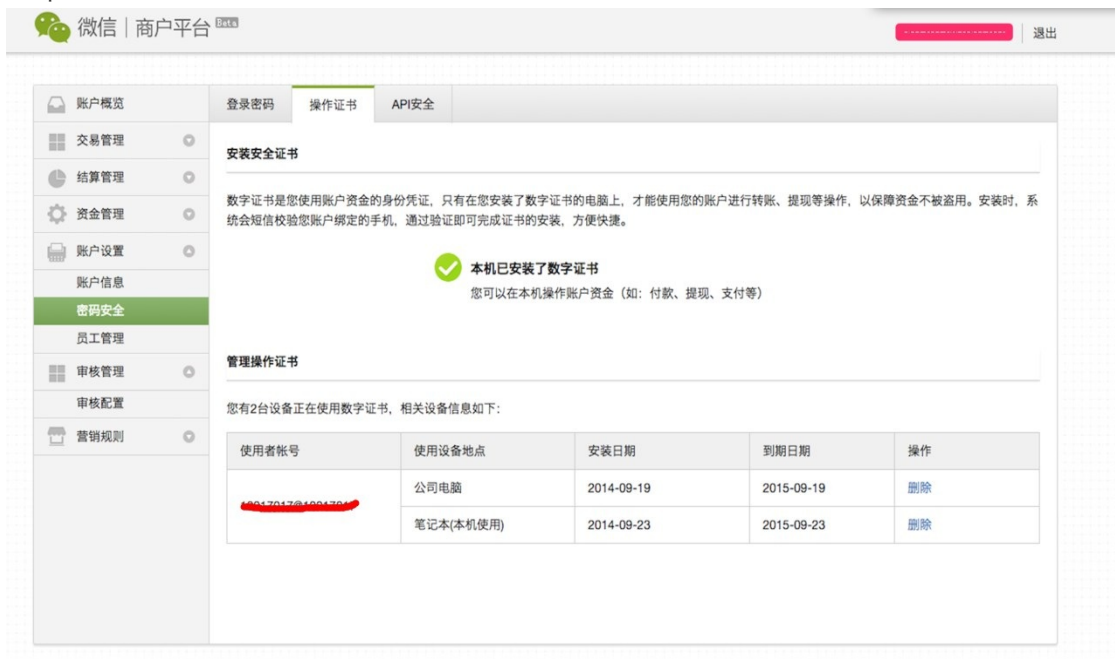


step 5 证书安装成功。



3 配置密钥。

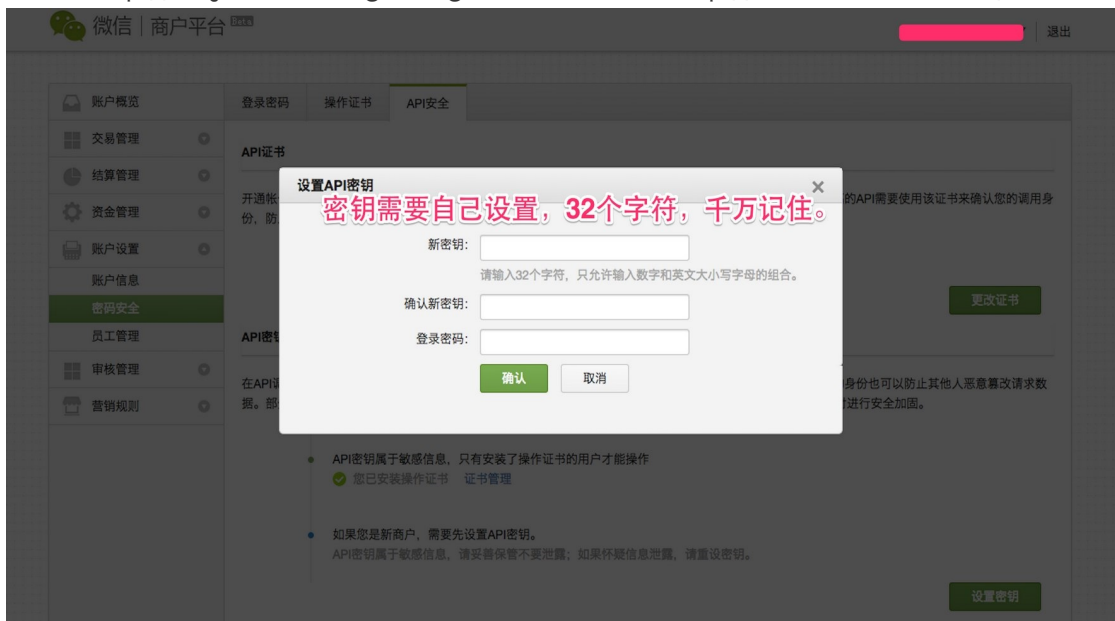
step1



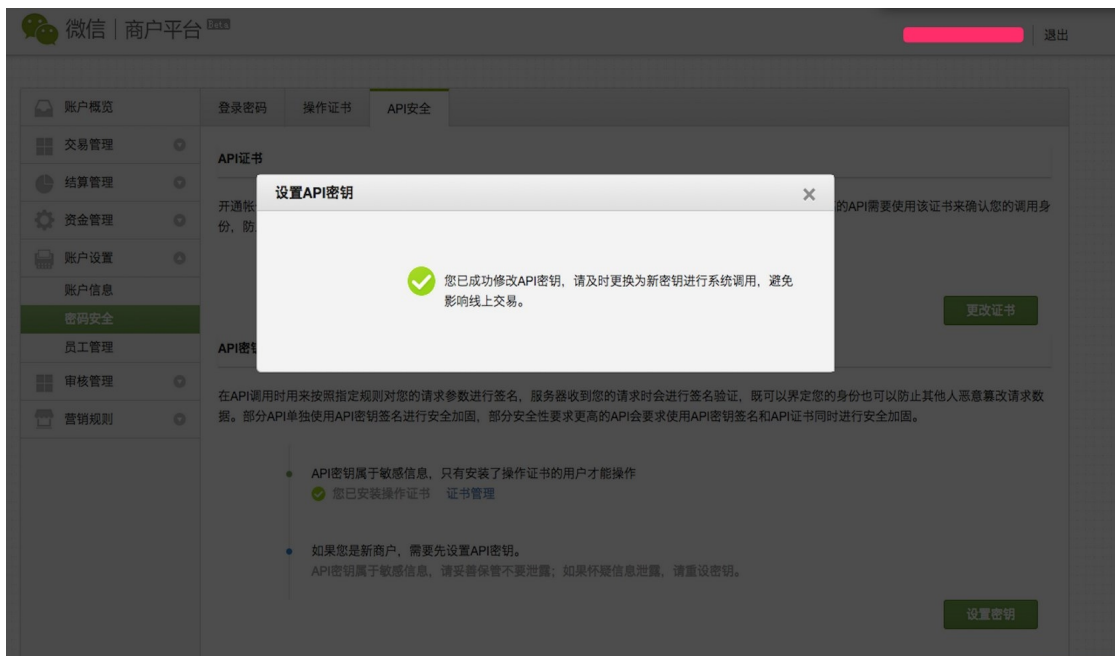
step2



step3 自己设置32位密钥，推荐使用在线密码生成器生成（个人自己写的无法发起支付），<http://suijimimashengcheng.51240.com> 或 <http://www.sexauth.com/>



step4



3.项目conf文件的配置

(1) 首先配置一下，ThinkPHP/Library/Org/WxPay/WxPayConfig.class.php文件，在里面配置APPID, MCHID, KEY, APPSECRET.

(2) 然后配置一下，Application/Common/Conf/config.php,里面配置下WxPayConf，里面需要填写MCHID,KEY, SSLCERT_PATH, SSLKEY_PATH, NOTIFY_URL

4.微信提供的sdk

微信提供的sdk，放在THINKPHP/Library/Org/WxPay文件夹下面，微信提供的是函数，目前全部封装到类里面了。

5. 微信支付的使用

(1) 步骤一，使用统一支付接口，获取prepay_id，并且把调用支付需要的参数jsApiParameters 返回给前台

```
//=====步骤1：使用统一支付接口，获取prepay_id=====
```

```
//使用统一支付接口
```

```
$unifiedOrder = new WxPayUnifiedOrder();
```

```
//设置统一支付接口参数
```

```
//设置必填参数
```

```
//appid已填,商户无需重复填写
```

```
//mch_id已填,商户无需重复填写
```

```
//noncestr已填,商户无需重复填写

//spbill_create_ip已填,商户无需重复填写

//sign已填,商户无需重复填写

$unifiedOrder->SetOpenid($openid); //用户标识

$unifiedOrder->SetBody("捷城科技充值");//商品描述

$unifiedOrder->SetOut_trade_no(md5($walletOrderId)); //商户订单号

$unifiedOrder->SetTotal_fee($amount * 100); //设置金额(把单位转化成分)

$unifiedOrder->SetNotify_url(C('WECHAT-HOST').C('WxPayConf.NOTIFY_URL'));//通知地址

$unifiedOrder->SetTrade_type("JSAPI");//交易类型


$order = WxPayApi::unifiedOrder($unifiedOrder);

$this->trackLog("order", $order);


if(!array_key_exists("result_code", $order) || $order['result_code'] != "SUCCESS"){

    //出现错误，把订单状态改为失败

    $walletOrderService->setStatusById($walletOrderId,
    WalletEnum::STATUS_ORDER_FAILED);

    $data['code'] = 0;

    $data['msg'] = "申请prepay_id失败！";

    $this->ajaxReturn($data);

}


$jsApi = new JsApiPay();

$jsApiParameters = $jsApi->GetJsApiParameters($order);
```

```
$this->trackLog('jsApiParameters',$jsApiParameters);
```

(2) 前台通过返回的参数调用微信支付，并且完成支付

```
function callpay(jsData){

    if (typeof WeixinJSBridge == "undefined"){

        if( document.addEventListener ){

            document.addEventListener("WeixinJSBridgeReady", jsApiCall, false);

        }else if (document.attachEvent){

            document.attachEvent("WeixinJSBridgeReady", jsApiCall);

            document.attachEvent("onWeixinJSBridgeReady", jsApiCall);

        }

    }else{

        jsApiCall(jsData);

    }

};

$.ajax({

    type: 'POST',

    dataType: 'json',

    url: APP_PATH+'/Company/Wallet/confirmRecharge',

    data: {

        "payType": 0,

        "amount": sVal

    },

    success: function (data) {

        if (data.code == 1) {
```

```

        // console.log(data.jsApiParameters);

        callpay(data.jsApiParameters);

    } else {

        $.toast(data.msg, 1000, "toast_orange");

    }

},

error: function () {

    $.toast("萝卜们太热情，服务器表示鸭梨很大！ ", 1000, "toast_orange");

}

});

```

(3) 后台收到微信的异步通知，获取支付结果并完成订单操作

```

$xml = file_get_contents("php://input"); //接收POST数据

//使用通用通知接口

$wxPayNotify = new WxPayNotify();

//存储微信的回掉

$notifyArray = $wxPayNotify->FromXml($xml);

$this->trackLog("$xml", $xml);

$this->trackLog("notifyArray", $notifyArray);

//验证签名，并回应微信。

//对后台通知交互时，如果微信收到商户的应答不是成功或超时，微信认为通知失败，

//微信会通过一定的策略（如30分钟共8次）定期重新发起通知，

```

//尽可能提高通知的成功率，但微信不保证通知最终能成功。

```
$wxPayNotify->Handle(true);
```

```
if(!array_key_exists("result_code", $notifyArray) || $notifyArray['result_code'] !=  
"SUCCESS"){
```

```
    $this->trackLog("notify result", "回调返回结果错误，直接返回，等待下次回调!");
```

```
    return;
```

```
}
```

具体的操作请看钱包的接口文档。