[微信小程序之wx.requestPayment 发起微信支付](https://www.cnblogs.com/jiqing9006/p/9318852.html)

wx.requestPayment 发起微信支付[#](https://www.cnblogs.com/jiqing9006/p/9318852.html" \l "2332848031)

timeStamp 时间戳

nonceStr 随机字符串

package 统一下单接口返回的 prepay\_id 参数值

signType 签名算法

paySign 支付签名

success 接口成功回调

fail 接口失败回调

complete 接口完成回调（成功，失败都执行）

1.先调用后台接口，生产基本数据

// 获取店铺信息

Api.BalancePay({

openid: openid,

amount: amount,

bid: bid,

}).then(res => {

if (res.errno) {

wx.showToast({ title: res.errdesc });

return;

}

var data = res.data;

...

})

具体的这个是做什么的呢？

判断用户是否存在。

判断充值id是否存在。

判断充值金额是否合法。

创建充值订单。

创建统一支付订单。

public function unifiedorder($openid,$order\_num,$total\_fee,$products\_name){

$trade\_no = $order\_num;

$url = 'https://api.mch.weixin.qq.com/pay/unifiedorder';

$data = [

'appid' => C('APPID'),

'mch\_id' => C('MCHID'),

'nonce\_str' => $this->createNonceStr(),

'sign\_type' => 'MD5',

'body' => $products\_name, //商品名称组合

'attach' => C('PAY\_ATTACH\_NAME'),

'out\_trade\_no' => $trade\_no, //订单号

'fee\_type' => 'CNY',

'total\_fee' => $total\_fee,

'spbill\_create\_ip' => $\_SERVER['REMOTE\_ADDR'],

'goods\_tag' => C('PAY\_ATTACH\_NAME'),

'notify\_url' => 'https://a.squmo.com/'.C('PAY\_URL\_NAME').'/Recharge/order\_notice',

'trade\_type' => 'JSAPI',

'openid' => $openid

];

// 获取签名

$sign = $this->MakeSign($data);

$data['sign'] = $sign;

$xml = $this->ToXml($data);

vendor('Func.Http');

// 提交获取数据

$result = $this->FromXml(Http::postXmlCurl($url,$xml));

return $result;

}

一切准备就绪之后，将一些数据返回给小程序。

$unifiedorder = $this->unifiedorder($openid,$order\_num,$total\_price,$products\_name);

$data = [

'appId' => C('APPID'),

'timeStamp' => time(),

'nonceStr' => $this->createNonceStr(),

'package' => 'prepay\_id='.$unifiedorder['prepay\_id'],

'signType' => 'MD5'

];

// 获取签名

$sign = $this->MakeSign($data);

$data['sign'] = $sign;

$this->json->setAttr('data',$data);

$this->json->Send();

2.再发起微信支付

wx.requestPayment({

'timeStamp': data.timeStamp.toString(),

'nonceStr': data.nonceStr,

'package': data.package,

'signType': 'MD5',

'paySign': data.sign,

'success': function (res) {

console.log('支付成功');

},

'fail': function (res) {

console.log('支付失败');

return;

},

'complete': function (res) {

console.log('支付完成');

var url = that.data.url;

console.log('get url', url)

if (res.errMsg == 'requestPayment:ok') {

wx.showModal({

title: '提示',

content: '充值成功'

});

if (url) {

setTimeout(function () {

wx.redirectTo({

url: '/pages' + url

});

}, 2000)

} else {

setTimeout(() => {

wx.navigateBack()

}, 2000)

}

}

return;

}

});

3.支付成功的回调（异步操作）

//微信支付回调

public function order\_notice(){

// 微信公众平台推送过来的post数据

$xml = $GLOBALS['HTTP\_RAW\_POST\_DATA'];

// 获取数据

$data = $this->FromXml($xml);

// 保存微信服务器返回的签名sign

$data\_sign = $data['sign'];

// sign不参与签名算法

unset($data['sign']);

$sign = $this->makeSign($data);

// 判断签名是否正确 判断支付状态

if ( ($sign===$data\_sign) && ($data['return\_code']=='SUCCESS') && ($data['result\_code']=='SUCCESS') ) {

$result = $data;

//获取服务器返回的数据

$order\_num = $data['out\_trade\_no']; //订单单号

$openid = $data['openid']; //付款人openID

$total\_fee = $data['total\_fee']; //付款金额

$transaction\_id = $data['transaction\_id']; //微信支付流水号

$user = M('user');

$user\_flag = $user->where(array('openid'=>$openid))->find();

$save\_data = array(

'total\_payed\_price' => $total\_fee, //实际到帐金额

'transaction\_id' => $transaction\_id,

'paytime' => time(),

'status' => 2 //1未支付;2已支付;3已申请退款;4已退款；5已完成

);

$recharge = M('recharge');

$recharge\_flag=$recharge->where(array('recoder'=>$order\_num,'uid'=>$user\_flag['id']))->find();

$recharge\_amount=$recharge\_flag['amount'];

$recharge\_save\_flag =$recharge->where(array('recoder'=>$order\_num,'uid'=>$user\_flag['id']))->save($save\_data);

if($recharge\_save\_flag){

$save\_balance['balance']= $user\_flag['balance']+$recharge\_amount;

$result\_balance =$user->where(array('openid'=>$openid))->save($save\_balance);

}

}else{

$result = false;

}

// 返回状态给微信服务器

if ($result) {

$str='<xml><return\_code><![CDATA[SUCCESS]]></return\_code><return\_msg><![CDATA[OK]]></return\_msg></xml>';

}else{

$str='<xml><return\_code><![CDATA[FAIL]]></return\_code><return\_msg><![CDATA[签名失败]]></return\_msg></xml>';

}

echo $str;

return $result;

}

其他常用函数

public function FromXml($xml)

{

if(!$xml){

throw new WxPayException("xml数据异常！");

}

//将XML转为array

//禁止引用外部xml实体

libxml\_disable\_entity\_loader(true);

$this->values = json\_decode(json\_encode(simplexml\_load\_string($xml, 'SimpleXMLElement', LIBXML\_NOCDATA)), true);

return $this->values;

}

public function ToXml($array){

if(!is\_array($array)|| count($array) <= 0){

return ;

}

$xml = '<xml version="1.0">';

foreach ($array as $key=>$val){

if (is\_numeric($val)){

$xml.="<".$key.">".$val."</".$key.">";

}else{

$xml.="<".$key."><![CDATA[".$val."]]></".$key.">";

}

}

$xml.="</xml>";

return $xml;

}

private function MakeSign($data)

{

//签名步骤一：按字典序排序参数

ksort($data);

$string = $this->ToUrlParams($data);

//签名步骤二：在string后加入KEY

$string = $string . "&key=".C('WEIXIN\_PAY\_KEY');

//签名步骤三：MD5加密

$string = md5($string);

//签名步骤四：所有字符转为大写

$result = strtoupper($string);

return $result;

}

private function ToUrlParams($array)

{

$buff = "";

foreach ($array as $k => $v)

{

if($k != "sign" && $v != "" && !is\_array($v)){

$buff .= $k . "=" . $v . "&";

}

}

$buff = trim($buff, "&");

return $buff;

}

作者：[TBHacker](http://www.cnblogs.com/jiqing9006)

出处：<https://www.cnblogs.com/jiqing9006/p/9318852.html>

版权：版权所有，转载注明作者和出处！