# 微信支付接口开发

## 1.支付模式：

1、公众号支付

公众号支付是用户在微信中打开商户的H5页面，商户在H5页面通过调用微信支付提供的JSAPI接口调起微信支付模块完成支付。应用场景有：

◆ 用户在微信公众账号内进入商家公众号，打开某个主页面，完成支付

◆ 用户的好友在朋友圈、聊天窗口等分享商家页面连接，用户点击链接打开商家页面，完成支付

◆ 将商户页面转换成二维码，用户扫描二维码后在微信浏览器中打开页面后完成支付

## 2.前期准备：

1. 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。

2. 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，填写后提交。信息如下图所示：



## 3.接口规则

### 1.协议规则：

商户接入微信支付，调用API必须遵循以下规则：



### 2.参数规则：

1、交易金额

交易金额默认为人民币交易，接口中参数支付金额单位为【分】，参数值不能带小数。对账单中的交易金额单位为【元】。

外币交易的支付金额精确到币种的最小单位，参数值不能带小数点。

2、交易类型

JSAPI--公众号支付、NATIVE--原生扫码支付、APP--app支付，统一下单接口trade\_type的传参可参考这里

MICROPAY--刷卡支付，刷卡支付有单独的支付接口，不调用统一下单接口

3、货币类型

货币类型的取值列表：

CNY：人民币

4、时间

标准北京时间，时区为东八区；如果商户的系统时间为非标准北京时间。参数值必须根据商户系统所在时区先换算成标准北京时间， 例如商户所在地为0时区的伦敦，当地时间为2014年11月11日0时0分0秒，换算成北京时间为2014年11月11日8时0分0秒。

5、时间戳

标准北京时间，时区为东八区，自1970年1月1日 0点0分0秒以来的秒数。注意：部分系统取到的值为毫秒级，需要转换成秒(10位数字)。

6、商户订单号

商户支付的订单号由商户自定义生成，微信支付要求商户订单号保持唯一性（建议根据当前系统时间加随机序列来生成订单号）。重新发起一笔支付要使用原订单号，避免重复支付；已支付过或已调用关单、撤销（请见后文的API列表）的订单号不能重新发起支付。

### 3.安全规则：

1、签名算法------（签名校验工具）

签名生成的通用步骤如下：

第一步，设所有发送或者接收到的数据为集合M，将集合M内非空参数值的参数按照参数名ASCII码从小到大排序（字典序），使用URL键值对的格式（即key1=value1&key2=value2…）拼接成字符串stringA。

特别注意以下重要规则：

◆ 参数名ASCII码从小到大排序（字典序）；

◆ 如果参数的值为空不参与签名；

◆ 参数名区分大小写；

◆ 验证调用返回或微信主动通知签名时，传送的sign参数不参与签名，将生成的签名与该sign值作校验。

◆ 微信接口可能增加字段，验证签名时必须支持增加的扩展字段

第二步，在stringA最后拼接上key得到stringSignTemp字符串，并对stringSignTemp进行MD5运算，再将得到的字符串所有字符转换为大写，得到sign值signValue。

key设置路径：微信商户平台(pay.weixin.qq.com)-->账户设置-->API安全-->密钥设置。

2、生成随机数算法

微信支付API接口协议中包含字段nonce\_str，主要保证签名不可预测。我们推荐生成随机数算法如下：调用随机数函数生成，将得到的值转换为字符串。

3、商户证书

（1）获取商户证书

微信支付接口中，涉及资金回滚的接口会使用到商户证书，包括退款、撤销接口。商家在申请微信支付成功后，收到的相应邮件后，可以按照指引下载API证书，也可以按照以下路径下载：微信商户平台(pay.weixin.qq.com)-->账户设置-->API安全-->证书下载 。

（2）使用商户证书

◆ apiclient\_cert.p12是商户证书文件，除PHP外的开发均使用此证书文件。

◆ 商户如果使用.NET环境开发，请确认Framework版本大于2.0，必须在操作系统上双击安装证书apiclient\_cert.p12后才能被正常调用。

◆ 商户证书调用或安装都需要使用到密码，该密码的值为微信商户号（mch\_id）

◆ PHP开发环境请使用商户证书文件apiclient\_cert.pem和apiclient\_key.pem ，rootca.pem是CA证书。

## 4.使用到的API：

### 1.统一下单：

（1）应用场景：除被扫支付场景以外，商户系统先调用该接口在微信支付服务后台生成预支付交易单，返回正确的预支付交易回话标识后再按扫码、JSAPI、APP等不同场景生成交易串调起支付。

（2）接口地址：<https://api.mch.weixin.qq.com/pay/unifiedorder>

### 2.查询订单：

1. 应用场景：该接口提供所有支付订单的查询，商户可以通过查询订单接口主动查询订单状态，完成下一步的业务。
2. 接口链接：https://api.mch.weixin.qq.com/pay/orderquery

### 3.关闭订单：

（1）应用场景：商户订单支付失败时需要生成新单号重新发起支付，要对原订单号调用关单，避免重复支付；系统下单后，用户支付超时，系统推出不在手里，避免用户继续，请调用关单接口。订单生成后不能马上调用关单接口，最短时间间隔是5分钟。

（2）接口链接：<https://api.mch.weixin.qq.com/pay/closeorder>

### 4.申请退款：

（1）应用场景：的那个交易发生后的一段时间内，由于买家和卖家的原因需要退款时，卖家可以通过退款接口将支付退款退还给买家，微信支付在将受到的退款请求并验收退款请求验证成功之后，按照退款规则将支付款按照原路退到买家的账号上面。

注意：

1.交易时间超过一年的时间订单将无法提交退款。

2.微信支付退款支持单笔交易分多次退款，多次退款需要提交原定单的商户订单号和设置不同的退款单号。一笔退款失败之后重新提交，要采用原来的退款订单号。总退款金额不能超过用户实际支付金额。

（2）接口地址：<https://api.mch.weixin.qq.com/secapi/pay/refund>

（3）需要证书。

### 5.支付结果通知：

（1）应用场景：支付完成后，微信会会把相关的支付结果和用户的信息发送给商户，商户需要接受处理，并返回答应。

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，微信认为通知失效，微信UI听过一定的策略定期重新发起通知，尽可能提高通知的成功率，但微信不保证通知最终能成功。

注意：

同样的通知可能会多次才能发给商户系统。商户系统必须能正确处理重复的通知。

1. 链接地址：

该链接是通过【统一下单API】中提交的参数notify\_url设置，如果链接无法访问，商户将无法接收到微信通知。

通知url必须为直接可访问的url，不能携带参数。示例：notify\_url：<https://pay.weixin.qq.com/wxpay/pay.action>

### 6微信相关接口的调用步骤：

#### 1.相关步骤：

1. 强请求实体转成数据集合。
2. 生成签名，并将签名添加到数据集合中。
3. 将数据集合转化成XML。
4. 发送post请求，并将返回信息转化成实体。

## 5.公众号支付：

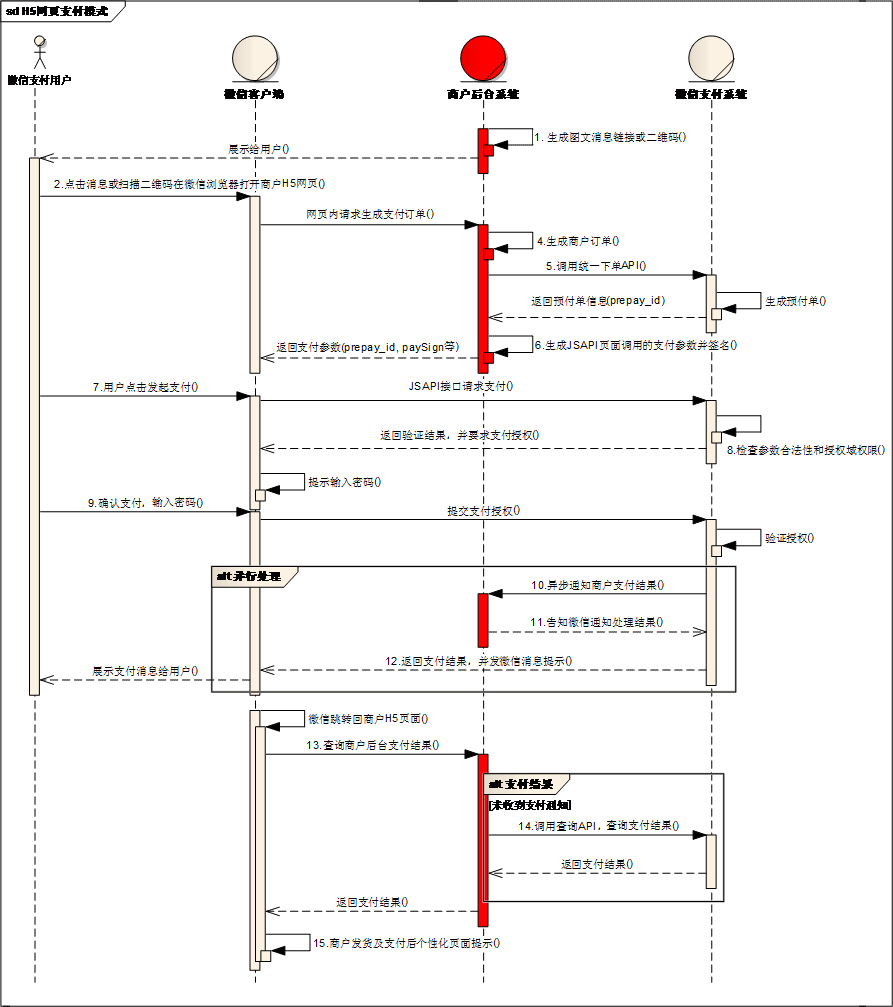
### 1.开发步骤：

（1）设置测试目录：在微信公众平台设置，支付测试状态下，设置测试目录，测试人的微信号添加到白名单，发起支付后的页面必须与设置的精确匹配。并将支付链接发到响应的公众号会话窗口才能正常发起支付测试。注意正式目录一定不能与测试目录设置成一样，否则支付会出错。



（2）设置正式支付目录：根据图中栏目书序进入修改栏目，勾选jsapi网页支付考通该权限，并配置好支付的授权目录，该目录必须是发起支付的页面的精确地目录，子目录下无法正常调用支付。

### 2.业务流程：

业务流程序列图：

## 公众号支付验收流程：

### 验收目的:

使用非充值代金券将调整商户的账单格式及API字段，需要制券商户与可用商户逐一完成以下用例，充分了解并适配账单格式与接口的更新后，方可使用非充值代金券产品； 同时非充值代金券优惠将造成可用商户订单实收金额的减少。

注意：

◆ 测试非充值代金券在沙箱环境发起的支付，非真实支付，不会产生实际的银行或零钱余额扣款。验收通过后，无需登录商户平台发起退款。

◆ 验收通过后，务必将”sandbox”关键词的url全部更换为正式的api链接，否则会影响商户正式交易。

◆ 刷卡支付，有单个用户当天支付5次或单笔金额500元以上需要输入支付密码的限制，因此，若客户端提示输入支付密码，请输入正确的密码完成支付流程。

### 2.验收流程:



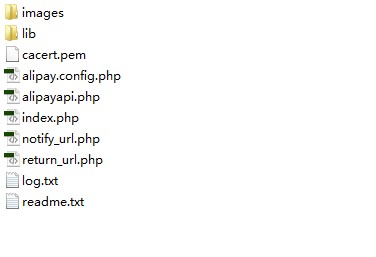
# 支付宝支付

## 下载支付宝接口包：

地址：https://doc.open.alipay.com/doc2/detail?treeId=62&articleId=103566&docType=1

## 整理接口文件：

（1）选择 create\_direct\_pay\_by\_user-PHP-UTF-8 这个名称的接口文件。里面包括如下文件：



（2）lib文件很重要，使整个接口的核心类文件：

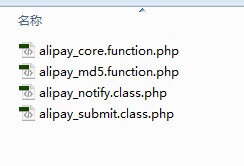
alipay.config.php是相关参数的配置文件

alipayapi.php 是支付宝接口入口文件

notify\_url.php 是服务器异步通知页面文件；

return\_url.php 是页面跳转同步通知文件；

（3）在thinkPHP的框架文件下，找到extend进入，在进入到vendor文件夹下，新建文件Ailpay，把支付宝作为第三方类文件引入。然后复制支付宝接口文件包里面的lib文件里面的全部文件。



（4）现在对以上文件进行重命名，

alipay\_core.function.php重命名为：Corefunction.php;

alipay\_md5.function.php重命名为：Md5function.php；

alipay\_notify.class.php重命名为：Notify.php；

alipay\_submit.class.php重命名为：Submit.php；

1. 打开submit.php文件，把代码删掉：

require\_once("alipay\_core.function.php");

require\_once("alipay\_md5.function.php");

1. 打来notify.php 文件，去掉代码：

require\_once("alipay\_core.function.php");

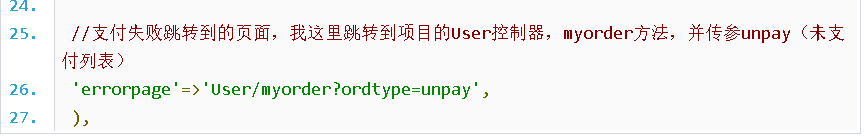
require\_once("alipay\_md5.function.php");

## 调用支付宝接口：

### 修改配置文件：

在文件conf/config.php文件中对支付宝的相关参数进行配置：





### 新建一个PayAction控制器代码如下：

















## 3.需要用到的函数：

Common/common.php中，这样不用手动调用，即可直接使用这些函数，代码如下：

//////////////////////////////////////////////////////

//Orderlist数据表，用于保存用户的购买订单记录；

/\* Orderlist数据表结构；

CREATE TABLE `tb\_orderlist` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`userid` int(11) DEFAULT NULL,购买者userid

`username` varchar(255) DEFAULT NULL,购买者姓名

`ordid` varchar(255) DEFAULT NULL,订单号

`ordtime` int(11) DEFAULT NULL,订单时间

`productid` int(11) DEFAULT NULL,产品ID

`ordtitle` varchar(255) DEFAULT NULL,订单标题

`ordbuynum` int(11) DEFAULT '0',购买数量

`ordprice` float(10,2) DEFAULT '0.00',产品单价

`ordfee` float(10,2) DEFAULT '0.00',订单总金额

`ordstatus` int(11) DEFAULT '0',订单状态

`payment\_type` varchar(255) DEFAULT NULL,支付类型

`payment\_trade\_no` varchar(255) DEFAULT NULL,支付接口交易号

`payment\_trade\_status` varchar(255) DEFAULT NULL,支付接口返回的交易状态

`payment\_notify\_id` varchar(255) DEFAULT NULL,

`payment\_notify\_time` varchar(255) DEFAULT NULL,

`payment\_buyer\_email` varchar(255) DEFAULT NULL,

`ordcode` varchar(255) DEFAULT NULL, //这个字段不需要的，大家看我西面的修正补充部分的说明！

`isused` int(11) DEFAULT '0',

`usetime` int(11) DEFAULT NULL,

`checkuser` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

\*/

//在线交易订单支付处理函数

//函数功能：根据支付接口传回的数据判断该订单是否已经支付成功；

//返回值：如果订单已经成功支付，返回true，否则返回false；

function checkorderstatus($ordid){

$Ord=M('Orderlist');

$ordstatus=$Ord->where('ordid='.$ordid)->getField('ordstatus');

if($ordstatus==1){

return true;

}else{

return false;

}

}

//处理订单函数

//更新订单状态，写入订单支付后返回的数据

function orderhandle($parameter){

$ordid=$parameter['out\_trade\_no'];

$data['payment\_trade\_no'] =$parameter['trade\_no'];

$data['payment\_trade\_status'] =$parameter['trade\_status'];

$data['payment\_notify\_id'] =$parameter['notify\_id'];

$data['payment\_notify\_time'] =$parameter['notify\_time'];

$data['payment\_buyer\_email'] =$parameter['buyer\_email'];

$data['ordstatus'] =1;

$Ord=M('Orderlist');

$Ord->where('ordid='.$ordid)->save($data);

}

/\*-----------------------------------

2013.8.13更正

下面这个函数，其实不需要，大家可以把他删掉，

具体看我下面的修正补充部分的说明

------------------------------------\*/

//获取一个随机且唯一的订单号；

function getordcode(){

$Ord=M('Orderlist');

$numbers = range (10,99);

shuffle ($numbers);

$code=array\_slice($numbers,0,4);

$ordcode=$code[0].$code[1].$code[2].$code[3];

$oldcode=$Ord->where("ordcode='".$ordcode."'")->getField('ordcode');

if($oldcode){

getordcode();

}else{

return $ordcode;

}

}

## 总结：

1、接口包中lib文件中的文件复制到Vendor后，重命名为TP规范的命名规则，为的是调用方便，当然你要改成其他名称也可以；

2、把执行支付操作（doalipay），处理异步返回结果（notifyurl），处理跳转返回结果（returnurl）三个支付接口的核心页面写到一个PayAction控制器中。

3、提交支付的页面中，可以在提交之前先把一些参数要传递的内容先通过隐藏域的方法组合好，比如金额先计算好，订单名称，订单描述等先用字符串组合好。然后提交表单，这样，在doalipay方法中只要直接构造传递参数，直接进行提交就行过了。

4、支付返回后的处理因为要在异步和跳转两个方法中都要进行相应的判断和处理，所以，把这些判断和处理写到一个自定义函数中，这样只要调用函数即可，使得代码更加清晰明了。

5、notify\_url和return\_url两种模式的返回url必须使用http://xxxxxxx这样的绝对路径，因为里是从支付宝平台返回到你的项目页面。不能使用相对路径。