# 微信支付申请开通教程

以下为无实体店铺的个人申请教程，约需成本600元以及一定时间，嫌麻烦者绕道

* 主要需准备：备案的域名 + 个体工商户营业执照（淘宝购买代办理约200-300元内） + 认证的微信小程序号（微信官方认证收取费用300元）

## 域名购买以及备案

* 进入阿里云或腾讯云等购买一个便宜的域名，关键需进行备案，详见各云平台备案流程

名字与后缀越不常见越便宜，例如腾讯云最低首年11元<https://buy.cloud.tencent.com/domain?domain=&tlds=&from=dnspodEntrance>

<https://cloud.tencent.com/product/ba>

<https://beian.aliyun.com/?spm=5176.12825654.amxosvpfn.19.7b652c4aSrV6hN>

拥有备案域名后才可获取微信支付Native与JSAPI接口

## 个体工商户营业执照

* 打开淘宝搜索 营业执照 即可，推荐找按销量高的且能提供注册地址的店铺，价格视各店铺地区而定，一般200-300元不等

提供注册地址说明：部分卖家可提供注册实体店铺地址，相当于借给你真实的地址挂靠以供注册详询淘宝卖家

说明：线下办理个体户营业执照是完全免费的，选择淘宝的代办主要是可提供注册地址以及跑腿费用，自行考虑。

**常见问题**

* 是否合法？ - 合法。详询淘宝卖家或银行工作人员。
* 有无后续其他费用？ - 无。详询淘宝卖家
* 年审怎么弄？ - 每年规定时间内登录 <http://www.gsxt.gov.cn/index.html> 选择 企业信息填报，自主填写年报进行年审即可。详询淘宝卖家
* 购买后等待3-5天至<http://www.gsxt.gov.cn/index.html>系统同步数据可查询到你的营业执照数据即可

## 微信认证小程序

微信支付需关联认证的应用以获取AppID账号：通过微信认证的服务号、政府或媒体类订阅号、小程序、APP、企业微信

<https://pay.weixin.qq.com/static/applyment_guide/applyment_index.shtml>

当然你有以下满足任一条件的应用也可以



推荐认证小程序账号是因为目前小程序认证仅需一次性费用300元，无需年审以及后续认证费用

* 打开 微信公众平台 注册企业小程序账号

<https://mp.weixin.qq.com/cgi-bin/registermidpage?action=index&lang=zh_CN&token=>

* 登录刚注册的小程序号开始申请认证目前费用为300元（申请过程中有个公函打印下来法人签字拍照上传即可）

<https://mp.weixin.qq.com/>

* 完成认证后记录小程序APPID 设置中账号信息查看



## 开始申请开通微信支付

准备好个体户营业执照与认证后的小程序账号后即可开始申请入驻

* 打开微信支付商户平台 点击 接入微信支付

<https://pay.weixin.qq.com/index.php/apply/applyment_home/guide_normal>

* 扫码登录填写注册普通商户申请单 经营场所填写PC网站和你的域名即可



* 等待审核开通成功

# 微信支付开发简介

## 开发准备

* 登入微信支付商户平台，点击产品中心添加认证的微信小程序的APPID



* 进入账户中心安装操作证书



* 紧接着进入左侧API安全设置秘钥Key（后续开发使用）



* 进入商户信息获取商户号mch\_id



## 开始开发

详见WechatController，填入你的配置 注意修改你的服务器回调链接



先生成订单号调用生成二维码，页面轮询再根据订单号查询的支付状态，官方也提供了notify\_url通知链接以供回调

/\*\*

\* 生成二维码

\* @param pay

\* @param request

\* @return

\* @throws AlipayApiException

\*/

@RequestMapping(value = "/wechat/precreate",method = RequestMethod.POST)

@ResponseBody

public Result<Object> getPayState(@ModelAttribute Pay pay, HttpServletRequest request) {

String wbody = "XPay-向作者Exrick捐赠";

String wnonce\_str = String.valueOf(System.currentTimeMillis());

BigDecimal wtotal\_fee = pay.getMoney().multiply(new BigDecimal("100.00")).setScale(0);

// 待签名字符串

String toSign = "appid="+appid+"&body="+wbody+"&mch\_id="+mch\_id+"&nonce\_str="+wnonce\_str+"&" +

"notify\_url="+notifyUrl+"&out\_trade\_no="+pay.getId()+"&spbill\_create\_ip="+ip+"&" +

"total\_fee="+wtotal\_fee+"&trade\_type=NATIVE&key="+key;

// 签名

String sign = WXPayUtil.MD5(toSign);

String body = "<xml>\n" +

" <appid>"+appid+"</appid>\n" +

" <mch\_id>"+mch\_id+"</mch\_id>\n" +

" <nonce\_str>"+wnonce\_str+"</nonce\_str>\n" +

" <body>"+wbody+"</body>\n" +

" <out\_trade\_no>"+pay.getId()+"</out\_trade\_no>\n" +

" <total\_fee>"+wtotal\_fee+"</total\_fee>\n" +

" <spbill\_create\_ip>"+ip+"</spbill\_create\_ip>\n" +

" <notify\_url>"+notifyUrl+"</notify\_url>\n" +

" <trade\_type>NATIVE</trade\_type>\n" +

" <sign>"+sign+"</sign>\n" +

"</xml>";

RestTemplate restTemplate = new RestTemplate();

restTemplate.getMessageConverters().set(1, new StringHttpMessageConverter(StandardCharsets.UTF\_8));

ResponseEntity<String> res = restTemplate.postForEntity("https://api.mch.weixin.qq.com/pay/unifiedorder", body, String.class);

Map<String, String> map = WXPayUtil.xmlToMap(res.getBody());

if("FAIL".equals(map.get("return\_code"))){

return new ResultUtil<Object>().setErrorMsg(map.get("return\_msg"));

}

Map<String, Object> result = new HashMap<>(16);

result.put("id", pay.getId());

result.put("qrCode", map.get("code\_url"));

return new ResultUtil<Object>().setData(result);

}

/\*\*

\* 查询支付结果

\* @param out\_trade\_no

\* @return

\* @throws AlipayApiException

\*/

@RequestMapping(value = "/wechat/query/{out\_trade\_no}",method = RequestMethod.GET)

@ResponseBody

public Result<Object> queryPayState(@PathVariable String out\_trade\_no) {

String wnonce\_str = String.valueOf(System.currentTimeMillis());

// 待签名字符串

String toSign = "appid="+appid+"&mch\_id="+mch\_id+"&nonce\_str="+wnonce\_str+"&out\_trade\_no="+out\_trade\_no+"&key="+key;

// 签名

String sign = WXPayUtil.MD5(toSign);

String body = "<xml>\n" +

" <appid>"+appid+"</appid>\n" +

" <mch\_id>"+mch\_id+"</mch\_id>\n" +

" <nonce\_str>"+wnonce\_str+"</nonce\_str>\n" +

" <out\_trade\_no>"+out\_trade\_no+"</out\_trade\_no>\n" +

" <sign>"+sign+"</sign>\n" +

"</xml>";

RestTemplate restTemplate = new RestTemplate();

restTemplate.getMessageConverters().set(1, new StringHttpMessageConverter(StandardCharsets.UTF\_8));

ResponseEntity<String> res = restTemplate.postForEntity("https://api.mch.weixin.qq.com/pay/orderquery", body, String.class);

Map<String, String> map = WXPayUtil.xmlToMap(res.getBody());

if("SUCCESS".equals(map.get("trade\_state"))){

// 支付成功回调

return new ResultUtil<Object>().setData(1);

}else{

return new ResultUtil<Object>().setData(0);

}

}

/\*\*

\* 微信通知回调

\* @return

\* @throws AlipayApiException

\*/

@RequestMapping(value = "/wechat/notify",method = RequestMethod.GET)

@ResponseBody

public String notify(HttpServletRequest request) {

try {

InputStream inStream = request.getInputStream();

int \_buffer\_size = 1024;

if (inStream != null) {

ByteArrayOutputStream outStream = new ByteArrayOutputStream();

byte[] tempBytes = new byte[\_buffer\_size];

int count = -1;

while ((count = inStream.read(tempBytes, 0, \_buffer\_size)) != -1) {

outStream.write(tempBytes, 0, count);

}

tempBytes = null;

outStream.flush();

//将流转换成字符串

String result = new String(outStream.toByteArray(), "UTF-8");

Map<String, String> map = WXPayUtil.xmlToMap(result);

if("SUCCESS".equals(map.get("result\_code"))){

// 支付成功回调

}

return "<xml>\n" +

"\n" +

" <return\_code><![CDATA[SUCCESS]]></return\_code>\n" +

" <return\_msg><![CDATA[OK]]></return\_msg>\n" +

"</xml>";

}

} catch (Exception e){

return null;

}

return null;

}

本demo通过前端轮询查询订单支付状态+官方通知回调+每日23:55再次检查订单，你也可以更加完善避免漏单，如增加定时任务、页面显示询问用户是否支付等