



链滴

# 微信支付 V3 版本的 Python 使用过程

作者: [YYJeffrey](#)

原文链接: <https://ld246.com/article/1627030646230>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



微信支付自推出V3版本接口以来，体验下来确实比V2有了很大的提升，但鉴于其官方没有提供Python版本的SDK，故在此记录通过Python3来使用微信支付V3的主要过程。

## 1. 搞清楚自己的身份

这一步看似可以忽略，但也需要仔细，笔者之前就犯了一个错，看了不对应的文档。

微信支付的商家有两种身份模式：**普通商户**、**服务商**。

一般的大部分商户都是属于普通商户，只有品牌商等才为服务商，并且两者身份不能互换。

**普通商户的API文档**：<https://pay.weixin.qq.com/wiki/doc/apiv3/index.shtml>

**服务商API文档**：[https://pay.weixin.qq.com/wiki/doc/apiv3\\_partner/index.shtml](https://pay.weixin.qq.com/wiki/doc/apiv3_partner/index.shtml)

## 2. 配置阶段

需要在商户端界面**申请API证书**和**配置APIv3密钥**，次过程可以参考每一个步骤右侧的查看指引按钮。



### 3.构造签名

V3版本的接口调用需要对请求进行签名，参考[https://pay.weixin.qq.com/wiki/doc/apiv3/wechatpay/wechatpay4\\_0.shtml](https://pay.weixin.qq.com/wiki/doc/apiv3/wechatpay/wechatpay4_0.shtml) 根据官方的文档可以按此步骤生成请求签名。

```
class WeixinPayUtil(object):
    @staticmethod
    def signature(private_key_path, sign_str):
        """
        生成签名值
        private_key_path 私钥路径
        sign_str 签名字符串
        https://pay.weixin.qq.com/wiki/doc/apiv3/wechatpay/wechatpay4_0.shtml
        """
        with open(private_key_path) as file:
            private_key = file.read()
        try:
            rsa_key = RSA.import_key(private_key)
            signer = pkcs1_15.new(rsa_key)
            digest = SHA256.new(sign_str.encode('utf-8'))
            return b64encode(signer.sign(digest)).decode('utf-8')
        except Exception:
            raise WeixinPaySignlError
```

```
def _generate_request_sign(self, url_path, data, method='POST'):
    """
    生成请求签名
    """
    sign_list = [method, url_path, self.timestamp, self.nonce_str]
    if data is not None:
        sign_list.append(json.dumps(data))
    else:
```

```

        sign_list.append("")
    sign_str = '\n'.join(sign_list) + '\n'
    return WeixinPayUtil.signature(private_key_path=self.cert_dir + 'api_key.pem', sign_str=sign_str)

```

## 统一支付

准备好了上面这一切就可以开始调用V3版本的接口了，以统一支付为例，笔者写了一个示例，其中部传参的命名和官方的命名稍有些不一致，至此整个V3版本的微信支付对接就告一段落了。

```

class WeixinPay(object):
    """
    微信支付
    """
    base_url = 'https://api.mch.weixin.qq.com'

    def __init__(self, mch_id, app_id, api_v3_key, mch_cert_no, cert_dir, notify_url):
        self.mch_id = mch_id
        self.app_id = app_id
        self.api_v3_key = api_v3_key
        self.mch_cert_no = mch_cert_no
        self.cert_dir = cert_dir
        self.notify_url = notify_url

        self.timestamp = str(int(time.time()))
        self.nonce_str = ''.join(random.sample(string.ascii_letters + string.digits, 16))

    def _generate_request_sign(self, url_path, data, method='POST'):
        """
        生成请求签名
        """
        sign_list = [method, url_path, self.timestamp, self.nonce_str]
        if data is not None:
            sign_list.append(json.dumps(data))
        else:
            sign_list.append("")
        sign_str = '\n'.join(sign_list) + '\n'
        return WeixinPayUtil.signature(private_key_path=self.cert_dir + 'api_key.pem', sign_str=sign_str)

    def _generate_pay_sign(self, app_id, package):
        """
        生成支付签名
        """
        sign_list = [app_id, self.timestamp, self.nonce_str, package]
        sign_str = '\n'.join(sign_list) + '\n'
        return WeixinPayUtil.signature(private_key_path=self.cert_dir + 'api_key.pem', sign_str=sign_str)

    def _generate_auth_header(self, signature):
        """
        生成授权请求头
        """
        return f'WECHATPAY2-SHA256-RSA2048 mchid="{self.mch_id}",nonce_str="{self.nonce_

```

```

tr}"," \
    f'signature="{signature}",timestamp="{self.timestamp}",serial_no="{self.mch_cert_no
"""

def unified_order(self, order_id, openid, amount, desc, mch_id=None, notify_url=None, prof
t_sharing=False,
    expire_time=None, attach=None, goods_tag=None, detail=None, scene_info=
one, currency='CNY'):
    """
    统一下单
    https://pay.weixin.qq.com/wiki/doc/apiv3/apis/chapter3_1_1.shtml
    """
    url_path = '/v3/pay/transactions/jsapi'
    url = self.base_url + url_path

    data = {
        'appid': self.app_id,
        'mchid': mch_id if mch_id is not None else self.mch_id,
        'description': desc,
        'out_trade_no': order_id,
        'notify_url': notify_url if notify_url is not None else self.notify_url,
        'settle_info': {
            'profit_sharing': profit_sharing
        },
        'amount': {
            'total': amount,
            'currency': currency
        },
        'payer': {
            'openid': openid
        }
    }

    if attach:
        data.update({'attach': attach})
    if expire_time:
        data.update({'time_expire': expire_time})
    if goods_tag:
        data.update({'goods_tag': goods_tag})
    if detail:
        data.update({'detail': detail})
    if scene_info:
        data.update({'scene_info': scene_info})

    signature = self._generate_request_sign(url_path=url_path, data=data)
    headers = {'Authorization': self._generate_auth_header(signature)}

    res = requests.post(url=url, json=data, headers=headers, timeout=TIMEOUT).json()

    # 支付签名
    pay_sign = self._generate_pay_sign(app_id=self.app_id, package='prepay_id=' + res['pre
ay_id'])

    return {

```

```
'timestamp': self.timestamp,  
'nonce_str': self.nonce_str,  
'prepay_id': res['prepay_id'],  
'sign_type': 'RSA',  
'pay_sign': pay_sign  
}
```