

微信公众号支付（JSAPI）对接备忘

2017-03-23 20:35 by buguge, 420 阅读, 评论, 收藏, 编辑

0 说明

本文里说的微信公众号支付对接指的是对接第三方支付平台的微信公众号支付接口。非微信支付官方文档里的公众号支付开发者文档那样的对接。不过，毕竟腾讯会把一部分渠道放给银行或有支付牌照的支付机构，所以，第三方的公众号支付也是在腾讯微信公众号支付的基础上做的改造。所以，基本的请求参数、签名机制、响应参数、交互流程、数据格式都一致的。

1 引言

.....

2 方案概述

2.1 行业背景

微信支付，是基于微信客户端提供的支付服务功能。同时向商户提供销售经营分析、账户和资金管理的功能支持。用户通过扫描二维码、反扫二维码等多种方式调起微信支付模块完成支付。

【二维码支付正扫和反扫的区别在哪里】

正扫：即收款码支付，也就是商户提供收款二维码，而消费者用手机APP扫码支付

反扫：即付款码支付，也就是消费者提供付款二维码，而商户使用扫描枪扫码收款

2.1.1 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

【两种实现方式】：原生态js支付和封装形式

2.2 业务实现流程

2.2.1 公众账号支付业务

微信内网页支付时序图

从图可知，商户系统涉及到的交互操作：

1. 生成图文消息链接或二维码
4. 生成商户订单
5. 调用统一下单API-----
6. 生成JSAPI页面调用的支付参数并签名
10. 异步通知商户结果-----
13. 查询后台支付结果
14. 调用查询API，查询支付结果-----

3 数据格式

xml

4 数字签名

4.2 签名算法

MD5签名

MD5是一种摘要生成算法，通过在签名原始串后加上商户通信密钥，进行MD5运算，形成的摘要字符串即为签名结果。

4.2.1 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。

5 补单机制

通知重试机制

6 公众账号支付接口

6.1 初始化请求接口

6.1.1 业务功能

初始化JSAPI 请求，通过生成token_id 来进行交互验证。

字段名	变量名	必填	类型	说明
是否原生态	is_raw	√	string(1)	是否原生态
用户openid	sub_openid	-	string(128)	微信用户关注商家公众号的openid（注：使用测试商户号不需要传用户openid； 切换正式的商户号需获取openid，并把获取的openid 值传给sub_openid。在切换成正式商户号传sub_openid 参数前，需提供正式商户号和公众号（服务号）appid 由渠道方配置，如果没有配置的话，会报sub_appid and su_openid not match 错误，导致无法正常调用接口）-----后文介绍如何获取openid
前台地址	callback_url	-	string(255)	交易完成后跳转的URL，需给绝对路径，255 字符内格式 如:http://wap.tenpay.com/callback.asp 注:该地址只作为前端页面的一个跳转，需使用notify_url 通知结果作为支付最终结果。-----扫码支付无此参数，即不需要商户上送交易完成的h5页面的。公众号支付这个参数可选，商户上送此页面也可做一些营销活动
是否限制信用卡	limit_credit_pay	-	String(32)	限定用户使用微信支付时能否使用信用卡，值为1，禁用信用卡；值为0 或者不传此参数则不禁用-----默认是支持信用卡的，所以此字段可不上送， 微信个人转账是不支持信用卡消费的，这正是与公众号支付和扫码支付最大的区别

几个特殊的请求字段

6.1.2 交互模式

请求：后台请求交互模式

返回结果+通知：后台请求交互模式+后台通知交互模式

注意：一般失败的结果不签名。

是否原生态is_raw 否String(1) 值为1：是（对应文档6.2 一节）；
值为0：否（对应文档6.3 一节）；不传默认是0

6.2 原生态js 支付接口-----原生态js支付实例：可以关注“1号外卖”这个公众号或者直接在手机微信里钱包中的手机充值查看

6.3 公众账号JS 支付接口-----这是所谓的“封装形式”（H5网页支付模式）

6.3.1 业务功能

初始化JSAPI 请求，通过生成token_id 来进行交互验证。

如调用时是用的原生态js 支付，此接口可以忽略

6.4 JS 支付通知接口

同《威富通扫码支付接口文档V1.4.2.pdf》6.2 扫码通知接口

—————【重要】首先必看.txt—————

1.开发时可以先使用测试商户号和密钥（demo中也都有写）

测试

商户号：'7551000001'

密钥：'9d101c97133837e13dde2d32a5054abb'

2.测试商户号有支付金额1元的限制，正式商户号不会有

3.文档中请求接口时传的参数，必填为是的参数是必须要传的（如有缺少会报错），必填为否的参数可以传也可以不传

4.返回参数中必填为是的参数是一定会返回的，必填为否的参数则不一定返回，必须以实际接收到的参数为准

5.注意看文档里提供了两种实现方式，PHP和C#版本demo中实现的是我们封装的形式，调用支付请求接口获取到token_id，然后组装成https://pay.swiftpass.cn/pay/jspay?token_id=9a0610bc519e782e6275e8c7dd94a445&showwxtitle=1这样的链接在服务号中调起支付（用户点击页面中的微信支付按钮时实际上就是点击的这个链接），JAVA版本则原生态js支付和封装形式都有实现，原生态支付时是调用支付请求接口获取到pay_info，取其中的参数去调用微信jsapi（这种方式最后调起微信时跟官方原生接口一致）
封装形式只需点击链接，简单些，实例：可以关注“广州市站”这个公众号具体看看
原生态js支付实例：可以关注“1号外卖”这个公众号或者直接在手机微信里钱包中的手机充值查看

6.demo中用测试商户号不需要传用户openid，即sub_openid参数置空
但切换正式的商户号时调用接口请求参数必须给sub_openid参数传入openid，同时请提供正式商户号和公众号（服务号）appid由我方配置，如果没有配置的话，会报sub_appid and su_openid not match错误，导致无法正常调用接口
获取openid 指导文档地址：http://www.cnblogs.com/twx1958/p/weixin76-user-info.html
获取公众号appid指导文档地址：http://jingyan.baidu.com/article/6525d4b12af618ac7c2e9468.html

—————公众号支付两种实现模式的说明—————

公众号支付流程是先调用接口文档6.1一节初始化请求接口获取到相应的返回值token_id和pay_info（对应的值是json 格式字符串，仅当is_raw=1时才返回），返回参数按 XML 的格式示例如下图：

然后下一个步骤分为了两个不同的模式：6.2一节原生态js支付和6.3一节公众账号JS支付

6.2原生态js支付

调用微信jspai方法，此方法只能在微信内置浏览器调用，在其他浏览器中无效，示例如下（注：示例代码中appid这六个参数有大小写，对应的值就是初始化请求接口中返回参数pay_info的值）：



```
function jsApiCall()
{
    WeixinJSBridge.invoke(
        'getBrandWCPayRequest', {
            "appId" : "wx1f87d44db95cba7a", //公众号名称, 由商户传入
            "timeStamp": "1468591622013", //时间戳, 自1970 年以来的秒数
            "nonceStr" : "1468591622013", //随机串
            "package" : "prepay_id=wx201607152207013ae4e376760784153308",
            "signType" : "MD5", //微信签名方式:
            "paySign" : "AD5A9E19D38002461812E09C0910A815" //微信签名,
        }, function(res){
            if(res.err_msg == "get_brand_wcpay_request:ok" ) {
                // 此处可以使用此方式判断前端返回, 微信团队郑重提示: res.err_msg 将在用户支付成功后返回ok, 但并不保证它绝对可靠。
                document.location.href="pay_success.jsp";
            }
            /* for(var i in res){
                alert(res[i]);
            } */
        });
}
```



最后在网页代码里调用这个js方法就可调起微信支付比如：

```
<button type="button" onclick="jsApiCall()" >微信支付</button>
```

但想正常弹出支付密码框，商户方开发人员必须提供自己的支付授权目录由服务商配置好，支付授权目录即为jspai方法代码所在页面的文件路径，如http://xxxx.com/zhifu/jsapi.html,授权目录就是http://xxxx.com/zhifu/（如果没有配置授权目录的话，支付时会无法弹出密码框或者提示当前页面URL未注册）

6.3 公众账号JS支付

这种模式是由我们进行了封装，用6.1一节初始化请求接口获取到的token_id值组装成https://pay.swiftpass.cn/pay/jspay?

token_id=1315838a57d0c83df0b62816220da070&showwxtitle=1这样的链接在手机微信中调起支付（用户点击页面中的微信支付按钮时实际上就是点击的这个链接，可以将链接放在手机微信文件传输助手去点击测试，注意token_id的值要更换有效的），这种模式不用实现jsapi方法，也不要提供支付授权目录。

对比6.2一节原生态js支付，直接点击链接的这个形式在支付弹出确认支付的弹出框时会多一个上面绿色下面白色背景页面，实际上这个背景页面是我们封装好的支付授权目录页面(对应固定的授权目录https://pay.swiftpass.cn/pay/)，这也是不用商户提供支付授权目录的原因所在。

效果如图：



总的来说，封装的链接形式开发时简单些，且无需商户提供自己的授权目录(由我们固定配置https://pay.swiftpass.cn/pay/)，但体验可能没有原生态js支付好。

参考：

微信支付开发文档 <https://pay.weixin.qq.com/wiki/doc/api/index.html>

微信支付公众账号支付开发文档 https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_1

微信公众号支付H5调用详解（附代码） http://blog.csdn.net/qq_28590639/article/details/49099275

获取微信用户的openid

关于用户openid：在关注者与公众号产生消息交互后，公众号可获得关注者的OpenID（加密后的微信号，每个用户对每个公众号的OpenID是唯一的。对于不同公众号，同一用户的openid不同）。

获取openid属于微信公众平台开发的范畴，可参考以下地址：<http://www.cnblogs.com/txw1958/p/weixin76-user-info.html> 第三节《通过OAuth2.0方式不弹出授权页面获得用户基本信息》

step1：配置回调域名

登陆微信公众平台，菜单“设置”→“公众号设置”→功能设置→网页授权域名



用户在网页授权页同意授权给公众号后，微信会将授权数据传给一个回调页面，回调页面需在此域名下，以确保安全可靠。

以上定义似乎会把人带入误区，我一开始被整懵了。配了个回调地址，其实不用，只需要配置授权访问的域名就ok了，这里我配置的是testpcenter.shenbianhui.cn。注意，要保证域名可访问并且要把MP_verify_****.txt放到站点相应的目录下，否则点击“确认”按钮会提示的。

step2：构造微信用户访问的url：

https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx8239ab824d12860&redirect_uri=http://testpcenter.shenbianhui.cn/QRCodeDemo/WeixinJSPay.aspx&response_type=code&scope=snsapi_base&state=1#wechat_redirect

其中，页面URL中的 scope=snsapi_base 表示应用授权作用域为不弹出授权页面，直接跳转。

这时，我们在redirect_uri指向的页面“<http://testpcenter.shenbianhui.cn/QRCodeDemo/WeixinJSPay.aspx>”程序里，就可获得get方式的两个参数值code和state。

step3：根据code获取openid：

请求url：[https://api.weixin.qq.com/sns/oauth2/access_token?](https://api.weixin.qq.com/sns/oauth2/access_token?appid=wx8639ab824d12861&secret=0e8d1234fd5345da5ea8e5fab61abcd7&code=02a9bed29b2185a9f0ed3a48fe56e700&grant_type=authorization_code)

appid=wx8639ab824d12861&secret=0e8d1234fd5345da5ea8e5fab61abcd7&code=02a9bed29b2185a9f0ed3a48fe56e700&grant_type=authorization_code

返回值: {"access_token":"k2iC-Bfce_1ukB1UffUnAB8AnFvGp_8_lvKiMTKF_hlLcjbKpFRrtmWJ5KeWvPOxEu6wsqvT4-oQzYyXVMM_sTTCBJycWupAOLdEXOtrM","expires_in":7200,"refresh_token":"g39EWv6L4FI7PV0859QPMw_VIMCVMCTco30Lk_-t35QP_mVhQjzvGIXk7MYk8nwkUsc-PxpT2a_kxcel5EAcwv41YizmdH7Hi7o57BlkKj4","openid":"o48_Ct5YigM7JDZ6x3Havr4kgzQQ","scope":"snsapi_base"}

注意, 二般情况下, 当code非法时, 返回的是{"errcode":40029,"errmsg":"invalid code, hints: [req_id: 1.Dnsa0402th10]"}

程序代码如下:

```

☐
public partial class WeixinJSPay : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        tbOpenId.Text = GetOpenIdByCode();
    }

    /// <summary>
    /// 微信用户访问时, 获取其openid
    /// </summary>
    private string GetOpenIdByCode()
    {
        /*
         * 给微信用户的请求地址:
         https://open.weixin.qq.com/connect/oauth2/authorize?
         appid=wx8239ab824d12860&redirect_uri=http://testpcenter.shenbianhui.cn/QRCodeDemo/WeixinJSPay.aspx&response_type=code&scope=snsapi_base&state=1#wechat
         */

        LogHelper.Write("-----:" + Request.Url); //示例: http://testpcenter.shenbianhui.cn/QRCodeDemo/WeixinJSPay.aspx?
        code=001bxIJx1PiIge0bZpLx1AAAJx1bxIJb&state=1

        string code = Request["code"];
        if (string.IsNullOrEmpty(code)) return "未获取到code参数";

        string url = string.Format(
            "https://api.weixin.qq.com/sns/oauth2/access_token?appid={0}&secret={1}&code={2}&grant_type=authorization_code",
            "wx8239ab824d12860",
            "0e8d4313fd5345da5ea8e5fab61ddae7",
            code);

        LogHelper.Write("-----请求openid: " + url);
        string resultJson = CommonModel.WebCommon.SendStreamStr("", url);
        LogHelper.Write("-----响应报文: " + resultJson);
        dynamic model = JsonConvert.DeserializeObject<dynamic>(resultJson);
        if (model.openid == null) return "未获取到openid";
        return model.openid;
    }
}
```