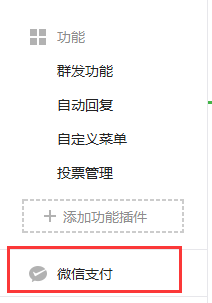
<https://www.cnblogs.com/qigege/p/5142366.html>

https://www.bbsmax.com/A/obzbBrLjzE/

进入微信公众号，点击功能菜单中的微信支付：并相应点击 使用教程-公众号支付



　　迅速对文档内容重温、浏览，以方便在Senparc.Weixin.MP.dll中查找相应的功能。



　　先配置支付授权目录，添加支付测试白名单，支付目录只支持三个，并且域名必须经过ICP备案。授权目录的作用是，如果要发起微信支付请求，请求 的链接地址必须在授权目录下，否则身份无效，支付不能成功。测试白名单中添加的个人微信号，才能完成微信支付测试目录支付的测试，不在白名单中人员发起支 付申请，支付不能成功。

　　配置完成后，如何调用呢？我们继续看官方说明：[H5调起支付API](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_7&index=6)

　　“在微信浏览器里面打开H5网页中执行JS调起支付。接口输入输出数据格式为JSON。

　　注意：WeixinJSBridge内置对象在其他浏览器中无效。

**列表中参数名区分大小，大小写错误签名验证会失败。”**

OK,这里说明了几个事情，第一必须在微信浏览器进行；第二，参数区分大小写；第三，数据格式为JSON。

　　官方说明，只要在页面中调用如下脚本，即可开启微信支付功能：

[复制代码](javascript:void(0);)

https://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

复制代码

1 function onBridgeReady(){

2 WeixinJSBridge.invoke(

3 'getBrandWCPayRequest', {

4 "appId" ： "wx2421b1c4370ec43b", //公众号名称，由商户传入

5 "timeStamp"：" 1395712654", //时间戳，自1970年以来的秒数

6 "nonceStr" ： "e61463f8efa94090b1f366cccfbbb444", //随机串

7 "package" ： "prepay\_id=u802345jgfjsdfgsdg888",

8 "signType" ： "MD5", //微信签名方式：

9 "paySign" ： "70EA570631E4BB79628FBCA90534C63FF7FADD89" //微信签名

10 },

11 function(res){

12 if(res.err\_msg == "get\_brand\_wcpay\_request：ok" ) {} // 使用以上方式判断前端返回,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。

13 }

14 );

15 }

16 if (typeof WeixinJSBridge == "undefined"){

17 if( document.addEventListener ){

18 document.addEventListener('WeixinJSBridgeReady', onBridgeReady, false);

19 }else if (document.attachEvent){

20 document.attachEvent('WeixinJSBridgeReady', onBridgeReady);

21 document.attachEvent('onWeixinJSBridgeReady', onBridgeReady);

22 }

23 }else{

24 onBridgeReady();

25 }

复制代码

[复制代码](javascript:void(0);)

　　我的调用代码：因为我要在点击按钮确认支付之后，在调用微信支付进行后续操作，把官方代码提出到方法中

[复制代码](javascript:void(0);)

复制代码

function onBridgeReady() {

WeixinJSBridge.invoke(

'getBrandWCPayRequest', {

"appId": $('#APPID').val(), //公众号名称，由商户传入

"timeStamp": $('#Timestamp').val(), //时间戳，自1970年以来的秒数

"nonceStr": $('#Noncestr').val(), //随机串

"package": $('#package').val(),

"signType": "MD5", //微信签名方式：

"paySign": $('#paySign').val() //微信签名

},

function (res) {

if (res.err\_msg == "get\_brand\_wcpay\_request:ok") {

//支付成功，后续自行处理

}

else

{

//支付取消，或者其他错误，自行处理

}

}

);

}

复制代码

[复制代码](javascript:void(0);)

　　好吧，那这堆参数是从哪来的，都是啥玩意儿？我们逐个分析一下：

　　appId：这个做微信开发都应该知道，公众号在开发者菜单就能找到

　　timeStamp：时间戳，官方描述为“自1970年以来的秒数”，不用担心，肯定能从支付类库里找到

　　nonceStr：官方解释是随机串“e61463f8efa94090b1f366cccfbbb444”，靠啥玩意儿？详见[随机数生成算法](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4_3)，原来就是一套加密规则和算法，做过URL请求接口的朋友应该知道，有些公司JSON串的签名方式和这比较类似。

　　package：预支付ID，调用官方API统一下单接口可以获得

　　signType：字符串"MD5"

　　paySign：官方解释是微信签名“70EA570631E4BB79628FBCA90534C63FF7FADD89”，好吧，我忍了，在看下[签名生成算法](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4_3)，看来和随机串一个鸟样

　　到这里，官方的接口说明已经了解的很清楚了，那么下面就要解决调用微信支付的这几个参数了，通过Senparc.Weixin.MP.dll应该如何使用呢？既然需要先调用统一下单接口获取预支付订单ID，好吧，我们先来研究一下，如何获得这个ID吧。

　　官方给出了[详细说明](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=9_1)，我们不在赘述，各参数研究按照上述接口的方式自行研究解决，唯一区别在于，调用官方接口需要传入一个XML，那很好办，拼接一下就可以了，预支付调用方法如下：

[复制代码](javascript:void(0);)

复制代码

//这里通过官方的一个实体，用户自行使用，我这里是直接读取的CONFIG文件

private static Senparc.Weixin.MP.TenPayLibV3.TenPayV3Info tenPayV3Info = new Senparc.Weixin.MP.TenPayLibV3.TenPayV3Info(ConfigurationManager.AppSettings["corpId"], ConfigurationManager.AppSettings["corpSecret"], ConfigurationManager.AppSettings["mch\_id"]

, ConfigurationManager.AppSettings["key"], ConfigurationManager.AppSettings["v3url"]);

/// <summary>

/// 微信预支付

/// </summary>

/// <param name="attach"></param>

/// <param name="body"></param>

/// <param name="openid"></param>

/// <param name="price"></param>

/// <param name="orderNum"></param>

/// <returns></returns>

public static string PayInfo(string attach, string body, string openid, string price, string orderNum = "1833431773763549")

{

RequestHandler requestHandler = new RequestHandler(HttpContext.Current);

//微信分配的公众账号ID（企业号corpid即为此appId）

requestHandler.SetParameter("appid", tenPayV3Info.AppId);

//附加数据，在查询API和支付通知中原样返回，该字段主要用于商户携带订单的自定义数据

requestHandler.SetParameter("attach", attach);

//商品或支付单简要描述

requestHandler.SetParameter("body", body);

//微信支付分配的商户号

requestHandler.SetParameter("mch\_id", tenPayV3Info.MchId);

//随机字符串，不长于32位。

requestHandler.SetParameter("nonce\_str", TenPayUtil.GetNoncestr());

//接收微信支付异步通知回调地址，通知url必须为直接可访问的url，不能携带参数。

requestHandler.SetParameter("notify\_url", tenPayV3Info.TenPayV3Notify);

//trade\_type=JSAPI，此参数必传，用户在商户公众号appid下的唯一标识。

requestHandler.SetParameter("openid", openid);

//商户系统内部的订单号,32个字符内、可包含字母，自己生成

requestHandler.SetParameter("out\_trade\_no", orderNum);

//APP和网页支付提交用户端ip，Native支付填调用微信支付API的机器IP。

requestHandler.SetParameter("spbill\_create\_ip", "127.0.0.1");

//订单总金额，单位为分，做过银联支付的朋友应该知道，代表金额为12位，末位分分

requestHandler.SetParameter("total\_fee", price);

//取值如下：JSAPI，NATIVE，APP，我们这里使用JSAPI

requestHandler.SetParameter("trade\_type", "JSAPI");

//设置KEY

requestHandler.SetKey(tenPayV3Info.Key);

requestHandler.CreateMd5Sign();

requestHandler.GetRequestURL();

requestHandler.CreateSHA1Sign();

string data = requestHandler.ParseXML();

requestHandler.GetDebugInfo();

//获取并返回预支付XML信息

return TenPayV3.Unifiedorder(data);

}

}

复制代码

[复制代码](javascript:void(0);)

好的，拿到预支付订单的返回数据，一切又都好办了，根据返回参数的不同，自行解决，我们只关心调用正确的过程，操作继续，在返回的正确XML数据 中，我们获取到了 <prepay\_id><! [CDATA[wx201411101639507cbf6ffd8b0779950874]]></prepay\_id>（官方示 例），好的，开始在页面做支付吧！

这里，我封装了一个实体，用来传输常用的数据，当然，各位也可以参考Senparc.Weixin.MP.dll提供的实体类。

[复制代码](javascript:void(0);)

复制代码

public class ShareInfo

{

string corpId = string.Empty;

public string CorpId

{

get { return corpId; }

set { corpId = value; }

}

string ticket = string.Empty;

public string Ticket

{

get { return ticket; }

set { ticket = value; }

}

string noncestr = string.Empty;

public string Noncestr

{

get { return noncestr; }

set { noncestr = value; }

}

string timestamp = string.Empty;

public string Timestamp

{

get { return timestamp; }

set { timestamp = value; }

}

private string paySign = string.Empty;

public string PaySign

{

get { return paySign; }

set { paySign = value; }

}

private string package = string.Empty;

public string Package

{

get { return package; }

set { package = value; }

}

}

复制代码

[复制代码](javascript:void(0);)

我们继续，来看一下支付接口需要用到的参数如何获取：

[复制代码](javascript:void(0);)

复制代码

public static ShareInfo GetPayInfo(string prepayid)

{

shareInfo = new ShareInfo();

//检查是否已经注册jssdk

if (!JsApiTicketContainer.CheckRegistered(corpId))

{

JsApiTicketContainer.Register(corpId, corpSecret);

}

JsApiTicketResult jsApiTicket = JsApiTicketContainer.GetTicketResult(corpId);

JSSDKHelper jssdkHelper = new JSSDKHelper();

shareInfo.Ticket = jsApiTicket.ticket;

shareInfo.CorpId = corpId.ToLower();

shareInfo.Noncestr = JSSDKHelper.GetNoncestr().ToLower();

shareInfo.Timestamp = JSSDKHelper.GetTimestamp().ToLower();

shareInfo.Package="prepay\_id=" + prepayid.ToLower();

RequestHandler requestHandler = new RequestHandler(HttpContext.Current);

requestHandler.SetParameter("appId", shareInfo.CorpId);

requestHandler.SetParameter("timeStamp", shareInfo.Timestamp);

requestHandler.SetParameter("nonceStr", shareInfo.Noncestr);

requestHandler.SetParameter("package", shareInfo.Package);

requestHandler.SetParameter("signType", "MD5");

requestHandler.SetKey(tenPayV3Info.Key);

requestHandler.CreateMd5Sign();

requestHandler.GetRequestURL();

requestHandler.CreateSHA1Sign();

shareInfo.PaySign = (requestHandler.GetAllParameters()["sign"]).ToString();

return shareInfo;

}

复制代码

[复制代码](javascript:void(0);)

这样，支付接口需要用到的参数，就都封装在ShareInfo里了，好吧，调用之后，我们回到页面的后置代码中，或者你采用的ORM对应代码中去，将参数输出到页面

[复制代码](javascript:void(0);)

https://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

复制代码

//处理页面支付调用信息

ShareInfo shareInfo = TenPayModule.GetPayInfo(prepayid);

System.Web.HttpContext.Current.Response.Write(string.Format("<input type=\"hidden\" id=\"Noncestr\" runat=\"server\" value=\"{0}\" />", shareInfo.Noncestr));

System.Web.HttpContext.Current.Response.Write(string.Format("<input type=\"hidden\" id=\"Timestamp\" runat=\"server\" value=\"{0}\" />", shareInfo.Timestamp));

System.Web.HttpContext.Current.Response.Write(string.Format("<input type=\"hidden\" id=\"APPID\" runat=\"server\" value=\"{0}\" />", shareInfo.CorpId));

System.Web.HttpContext.Current.Response.Write(string.Format("<input type=\"hidden\" id=\"paySign\" runat=\"server\" value=\"{0}\" />", shareInfo.PaySign));

System.Web.HttpContext.Current.Response.Write(string.Format("<input type=\"hidden\" id=\"package\" runat=\"server\" value=\"{0}\" />", shareInfo.Package));

复制代码

[复制代码](javascript:void(0);)

好的，写到这里，大家参照上面的JS代码，就可以完成整个的支付功能了。最后，再附送一个生成商家订单号的方法，代码如下：

[复制代码](javascript:void(0);)

https://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

复制代码

public string GetOrderNumber()

{

string Number = DateTime.Now.ToString("yyMMddHHmmss");

return Number + Next(1000, 1).ToString();

}

private static int Next(int numSeeds, int length)

{

byte[] buffer = new byte[length];

System.Security.Cryptography.RNGCryptoServiceProvider Gen = new System.Security.Cryptography.RNGCryptoServiceProvider();

Gen.GetBytes(buffer);

uint randomResult = 0x0;

for (int i = 0; i < length; i++)

{

randomResult |= ((uint)buffer[i] << ((length - 1 - i) \* 8));

}

return (int)(randomResult % numSeeds);

}

复制代码

[复制代码](javascript:void(0);)

好了，一切都是这样的简单，一两个小时，微信支付轻松搞定！