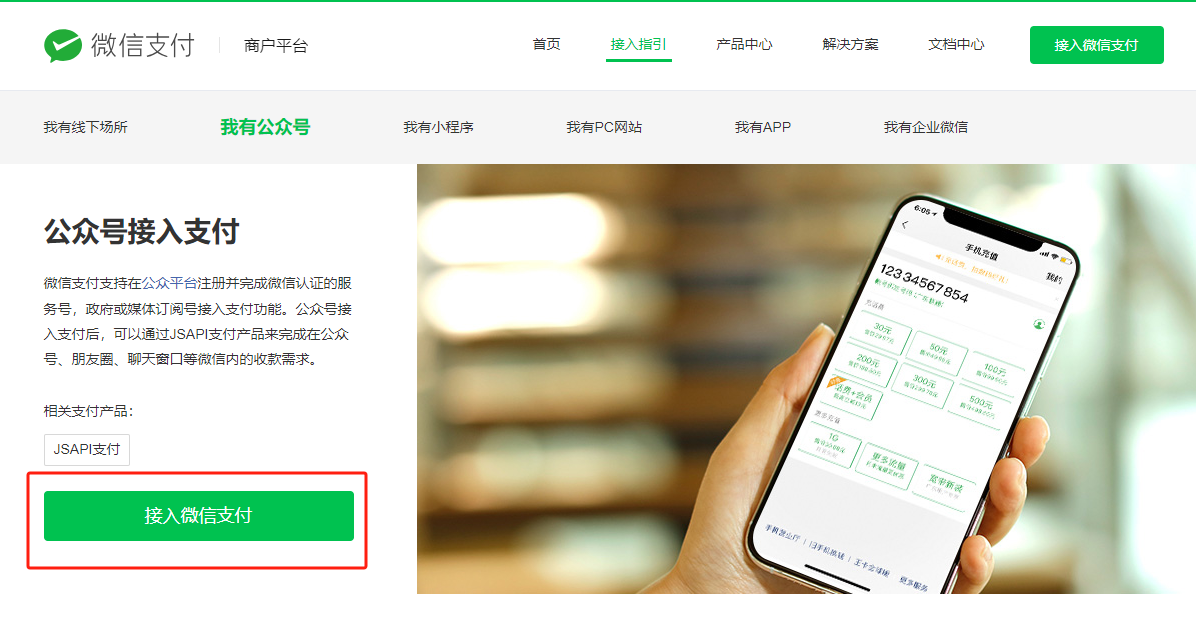
## 申请微信商户

申请地址：<https://pay.weixin.qq.com/static/applyment_guide/applyment_detail_public.shtml>



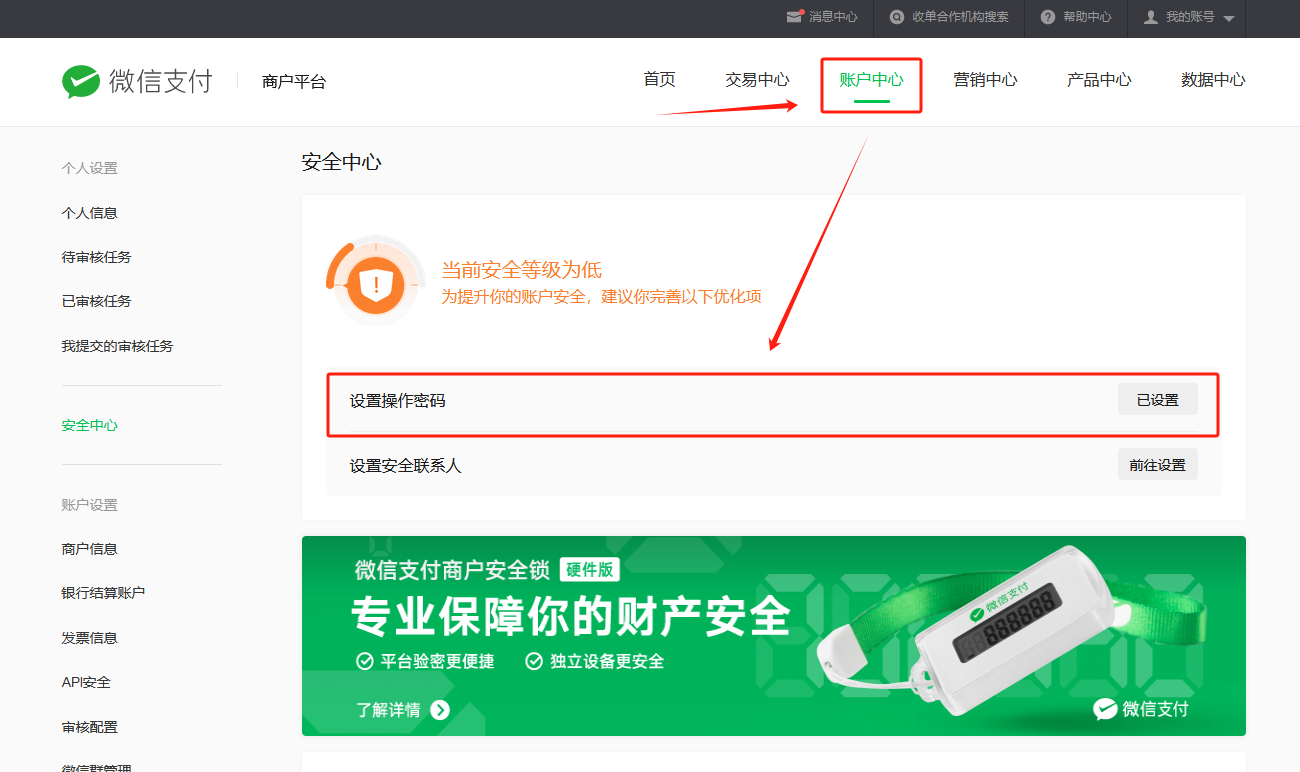


**所需材料：**

* 营业执照
* 法人身份证正/反面照片
* 经营行业信息
* 结算账户

## 支付相关配置

1. **设置操作密码**



1. **API安全配置**



**“申请API证书”**：根据网站指引去申请即可，证书用于程序发起支付相关的接口，具体如何使用看第3节。

**“设置APIv3密钥”**：根据网站指引去设置即可，密钥用于调用支付相关接口，具体如何使用看第3节。（腾讯坑：如果新申请商户，第一次设置密钥后，开发发起生成预支付订单请求可能会报“签名错误”，可以尝试来重置密钥再请求试试，本人就是重置解决了“签名错误”的问题，网络上也很多人也有这个反馈）

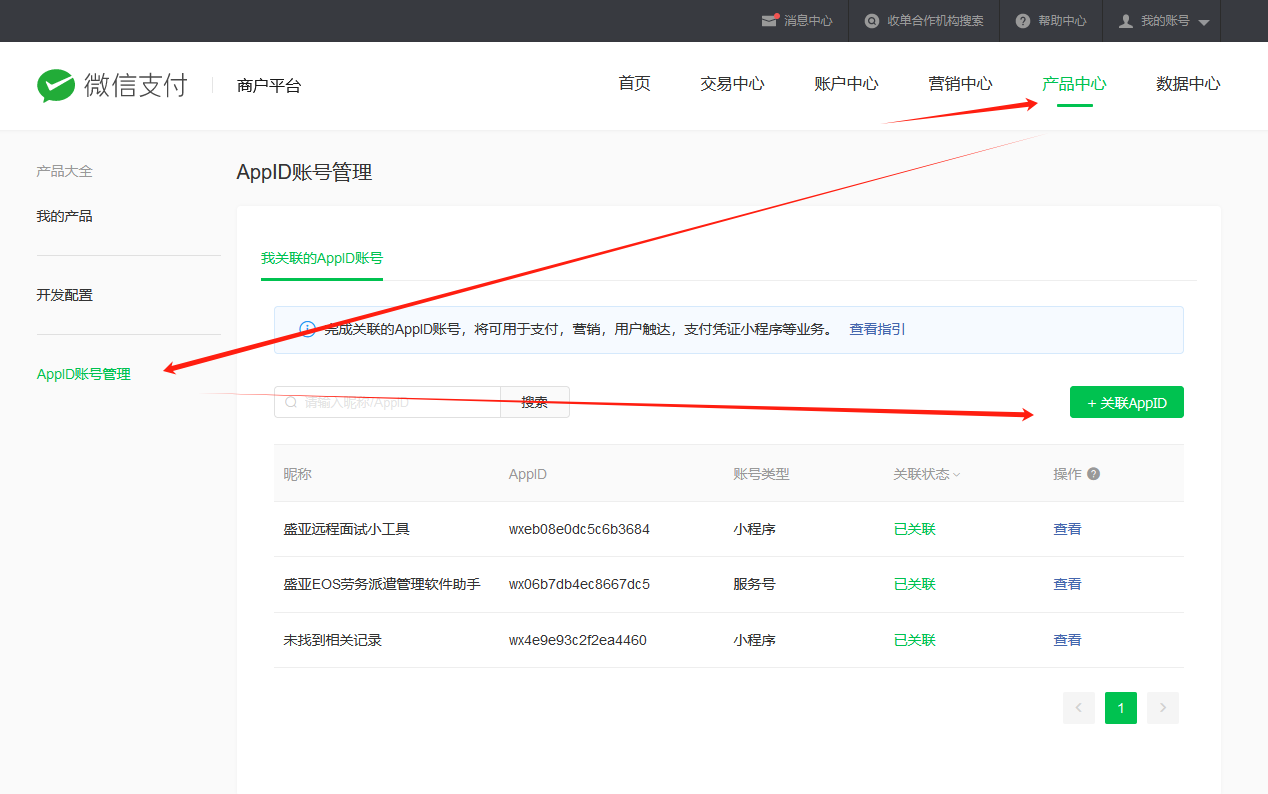
1. **开发配置**

主要是添加JSAPI支付授权目录(即网站域名)



1. **AppID账号管理**

关联需要开发微信支付的平台(公众号、小程序)



注意：

这里发起关联后，还需要小程序或公众号管理员确认（微信会给管理员发送微信信息，管理员手机上点微信信息可以确认关联，如果管理员没收到微信消息，可以去公众号后台确认：广告与服务->微信支付）。一定要确保这里的“关联状态”是“已关联”

理论上只能关联通主体的公众号和小程序，但好像也可以关联非同主体的，只要发起关联的时候签署“联合运营”协议即可，我还没实际操作过。

## 代码实现

我的实现技术方案是.netcore3.1 + 盛派组件。可以参考高层次人才项目，或者盛派的源码Senparc.Weixin.Sample.NetCore3->WeiXin->TenPayV3Controller.cs。

高层次人才项目关键代码如下：

1. **Startup.cs -> ConfigureServices 方法里注册盛派服务：**

//Senparc.Weixin 注册（必须）

services.AddSenparcWeixinServices(\_appConfiguration);

services.AddSenparcGlobalServices(\_appConfiguration)//Senparc.CO2NET 全局注册

.AddSenparcWeixinServices(\_appConfiguration);//Senparc.Weixin 注册（如果使用Senparc.Weixin SDK则添加）



1. **Startup.cs -> Configure 方法里配置盛派服务：**

//Senparc

var senparcSetting = \_senparcSetting.Value;

var senparcWeixinSetting = \_senparcWeixinSetting.Value;

IRegisterService register = RegisterService.Start(senparcSetting).UseSenparcGlobal();// 启动 CO2NET 全局注册，必须！

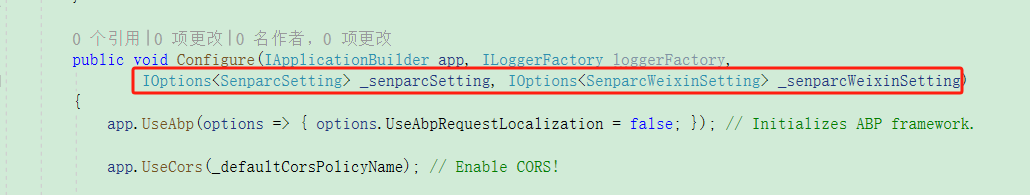
register.ChangeDefaultCacheNamespace("DefaultCO2NETCache");

register.RegisterTraceLog(ConfigTraceLog);

register.UseSenparcWeixin(senparcWeixinSetting, senparcSetting);//微信全局注册，必须！

register.RegisterTenpayV3(senparcWeixinSetting, "上海盛亚信息技术有限公司");

Senparc.Weixin.Config.ThrownWhenJsonResultFaild = false; //当 JsonResult 不为“成功”状态时，是否抛出异常，默认为 true



1. **appsettings.json配置：**



注意：

支付证书放到服务器上要记得安装（windows服务器双击即可安装）；支付证书密码默认是商户号；

1. **生成预支付订单并组装微信支付参数给页面**

public ActionResult JsApi()

{

try

{

var openId = \_memoryCache.Get<string>("OpenId");//进行登录

if (string.IsNullOrEmpty(openId))

return Content("未获取到openid");

string sp\_billno = Request.Query["order\_no"];

if (string.IsNullOrEmpty(sp\_billno))

{

//生成订单10位序列号，此处用时间和随机数生成，商户根据自己调整，保证唯一

sp\_billno = string.Format("{0}{1}{2}", TenPayV3Info.MchId /\*10位\*/, SystemTime.Now.ToString("yyyyMMddHHmmss"),

TenPayV3Util.BuildRandomStr(6));

//注意：以上订单号仅作为演示使用，如果访问量比较大，建议增加订单流水号的去重检查。

}

else

{

sp\_billno = Request.Query["order\_no"];

}

var timeStamp = TenPayV3Util.GetTimestamp();

var nonceStr = TenPayV3Util.GetNoncestr();

var body = "TestProduct";

var price = 1;//单位：分

var xmlDataInfo = new TenPayV3UnifiedorderRequestData(TenPayV3Info.AppId, TenPayV3Info.MchId, body, sp\_billno, price, HttpContext.UserHostAddress()?.ToString(), TenPayV3Info.TenPayV3Notify, TenPayV3Type.JSAPI, openId, TenPayV3Info.Key, nonceStr);

//var s = xmlDataInfo.PackageRequestHandler.ParseXML();//获取XML;

//return Content(s);

var result = TenPayV3.Unifiedorder(xmlDataInfo);//调用统一订单接口

//JsSdkUiPackage jsPackage = new JsSdkUiPackage(TenPayV3Info.AppId, timeStamp, nonceStr,);

//return Content(Newtonsoft.Json.JsonConvert.SerializeObject(result));

var package = string.Format("prepay\_id={0}", result.prepay\_id);

//ViewData["product"] = product;

ViewData["appId"] = TenPayV3Info.AppId;

ViewData["timeStamp"] = timeStamp;

ViewData["nonceStr"] = nonceStr;

ViewData["package"] = package;

ViewData["paySign"] = TenPayV3.GetJsPaySign(TenPayV3Info.AppId, timeStamp, nonceStr, package, TenPayV3Info.Key);

//临时记录订单信息，留给退款申请接口测试使用

\_memoryCache.Set("BillNo", sp\_billno);

\_memoryCache.Set("BillFee", price.ToString());

return View();

}

catch (Exception ex)

{

var msg = ex.Message;

msg += "<br>" + ex.StackTrace;

msg += "<br>==Source==<br>" + ex.Source;

if (ex.InnerException != null)

{

msg += "<br>===InnerException===<br>" + ex.InnerException.Message;

}

return Content(msg);

}

}

1. **页面JSAPI发起支付请求**

@\*DPBMARK\_FILE TenPay\*@

@\*提示：当前过程模拟的是订单支付过程（订单已经生成），不能对同一个订单号重复支付 \*@

@{ Layout = null; }

<!DOCTYPE html>

<html>

<head>

<title>公众号jsapi支付测试网页</title>

<meta http-equiv="Content-Type" content="text/html; charset=GBK" />

<script src="~/lib/jquery/dist/jquery.js"></script>

<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1" />

<style>

body {

margin: 0;

padding: 0;

background: #eae9e6;

}

body, p, table, td, th {

font-size: 14px;

font-family: helvetica,Arial,Tahoma;

}

h1 {

font-family: Baskerville,HelveticaNeue-Bold,helvetica,Arial,Tahoma;

}

a {

text-decoration: none;

color: #385487;

}

.product {

font-size: 200%;

color: #a5a5a5;

}

.title h1 {

font-size: 22px;

font-weight: bold;

padding: 0;

margin: 0;

line-height: 1.2;

color: #1f1f1f;

}

.title p {

color: #aaa;

font-size: 12px;

margin: 5px 0 0;

padding: 0;

font-weight: bold;

}

.WCPay {

padding-top: 3%;

}

</style>

<script>

function auto\_remove(img) {

div = img.parentNode.parentNode; div.parentNode.removeChild(div);

img.onerror = "";

return true;

}

function changefont(fontsize) {

if (fontsize < 1 || fontsize > 4) return;

$('#content').removeClass().addClass('fontSize' + fontsize);

}

// 当微信内置浏览器完成内部初始化后会触发WeixinJSBridgeReady事件。

document.addEventListener('WeixinJSBridgeReady', function onBridgeReady() {

//公众号支付

jQuery('a#getBrandWCPayRequest').click(function (e) {

WeixinJSBridge.invoke('getBrandWCPayRequest', {

"appId": "@ViewData["appId"]", //公众号名称，由商户传入

"timeStamp": "@ViewData["timeStamp"]", //时间戳

"nonceStr": "@ViewData["nonceStr"]", //随机串

"package": "@Html.Raw(ViewData["package"])",//扩展包

"signType": "MD5", //微信签名方式:MD5

"paySign": "@ViewData["paySign"]" //微信签名

}, function (res) {

//alert(JSON.stringify(res));

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

if (confirm('支付成功！点击“确定”进入退款流程测试。')) {

location.href = '@Url.Action("Refund", "TenPayV3")';

}

//console.log(JSON.stringify(res));

}

// 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。

//因此微信团队建议，当收到ok返回时，向商户后台询问是否收到交易成功的通知，若收到通知，前端展示交易成功的界面；若此时未收到通知，商户后台主动调用查询订单接口，查询订单的当前状态，并反馈给前端展示相应的界面。

});

});

WeixinJSBridge.log('yo~ ready.');

}, false);

if (jQuery) {

jQuery(function () {

var width = jQuery('body').width() \* 0.87;

jQuery('img').error(function () {

var self = jQuery(this);

var org = self.attr('data-original1');

self.attr("src", org);

self.error(function () {

auto\_remove(this);

});

});

jQuery('img').each(function () {

var self = jQuery(this);

var w = self.css('width');

var h = self.css('height');

w = w.replace('px', '');

h = h.replace('px', '');

if (w <= width) {

return;

}

var new\_w = width;

var new\_h = Math.round(h \* width / w);

self.css({ 'width': new\_w + 'px', 'height': new\_h + 'px' });

self.parents('div.pic').css({ 'width': new\_w + 'px', 'height': new\_h + 'px' });

});

});

}

</script>

</head>

<body>

<div class="WCPay">

<a id="getBrandWCPayRequest" href="javascript:void(0);">

<div class="product">

您已选中产品：test<br />

单价：1分

</div>

<h1 class="title">点击提交可体验微信支付</h1>

<p>提示：支付成功后可退款，并体验退款流程</p>

</a>

</div>

</body>

</html>