<https://pay.weixin.qq.com/wiki/doc/api/index.html>

<https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_1>

1. 申请微信公众号
2. 申请微信支付
3. 通过后再商户界面设置key
4. 注册微信js-sdk
5. 后台调用微信统一下单
   1. 签名重点
6. 前台拿到签名和微信单号
   1. 利用js-sdk 吊起微信支付
   2. 吊起的时候如果失败，可能是当前URL为被微信支付授权；
7. 微信支付结果处理：
   1. 编写后台异步通知接口==根据微信返回值做业务处理【同样需要签名】
   2. 编写前台跳转代码

注意：

            前台调试时打开微信js-sdk的debug模式=true

====================页面代码：=============

<script src="<http://res.wx.qq.com/open/js/jweixin-1.0.0.js>" type="text/javascript"></script>

**function**pay(){

        $(".pay-loading").show();

        $.ajax({

            type: "Post",

            url: "${ctx}/pay/getprepayid\_appreciation",

            dataType: "json",

            data:{"orderId":"${appreciationOrder.id?c}"},

            success: **function**(data){

                $(".pay-loading").hide();

**if**(data.flag==**true**){

                    wx.chooseWXPay({

                        timestamp:data.timeStamp,         //时间戳，自1970年以来的秒数

                        nonceStr: data.nonceStr, //随机串

**package**: data.package,

                        signType: data.signType,         //微信签名方式：

                        paySign: data.paySign, //微信签名

                        success: **function**(res) {

                            // 支付成功后的回调函数

                            alert("支付成功");

                            $(".pay-loading").show();

                            window.location.href="${ctx}/pay/payconfirm\_appreciation";

                        },

                        fail: **function**(res) {

                            //接口调用失败时执行的回调函数。

                            alert("支付失败,请重试!");

                        },complete: **function**(res) {

                            //接口调用完成时执行的回调函数，无论成功或失败都会执行。

//                            alert(res);

                        },cancel: **function**(res) {

                            //用户点击取消时的回调函数，仅部分有用户取消操作的api才会用到。

//                            alert(res);

                        },trigger: **function**(res) {

                            //监听Menu中的按钮点击时触发的方法，该方法仅支持Menu中的相关接口。

//                            alert(res);

                        }

                    });

                }**else**{

                    alert(data.msg);

                }

            }

        });

    }

====================后台代码：==============

*/\*\**

*\* 增项付款使用-微信预支付统一下单*

*\*****@param****request*

*\*****@param****orderId*

*\*****@return***

*\*/*

@RequestMapping("getprepayid\_appreciation")

    @ResponseBody

    public Map<String,Object> getPrepayidAppreciation(HttpServletRequest request,Long orderId) {

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

        try {

            Long customerId = (Long)request.getSession().getAttribute("customerId");

            CustomerVo customerVo =  customerService.selectByPrimaryKey(customerId).getResult();

            ServiceResult<AppreciationOrderVo> queryResult = appreciationOrderService.selectByPrimaryKey(orderId);

            if(queryResult.getSuccess()){

                AppreciationOrderVo orderVo = queryResult.getResult();

//                封装有序参数,微信签名使用

                TreeMap<String,String> treeMap = new TreeMap<>();

                treeMap.put("appid",Configure.*getAppid*());

                treeMap.put("mch\_id",Configure.*getMchid*());

                treeMap.put("device\_info","weixin");

                treeMap.put("body",orderVo.getServiceProductName());

                treeMap.put("nonce\_str",RandomStringGenerator.*getRandomStringByLength*(32));

                treeMap.put("attach",orderVo.getServiceNetworkName());

                treeMap.put("out\_trade\_no",orderVo.getOrderNo()+"");

                treeMap.put("total\_fee","1");//金额生产环境需改为动态获取

                treeMap.put("spbill\_create\_ip",IpUtil.*getIpAddr*(request));

                treeMap.put("trade\_type","JSAPI");

                treeMap.put("notify\_url",Configure.*notify\_url\_order\_appreciation*);

                treeMap.put("openid",customerVo.getOpenId());

//解析有序参数集合 拼接成加密字符串

                Set<String> keys = treeMap.keySet();

                StringBuffer params = new StringBuffer();

                for (String key : keys){

                    String value = treeMap.get(key);

                    params.append(key);

                    params.append("=");

                    params.append(value);

                    params.append("&");

                }

                params.append("key="+Configure.*getKeyPay*());

                String a = params.toString();

                System.*out*.println("a=" + a);

//                获取加密签名

                String sign =  MD5.*MD5Encode*(a).toUpperCase();

                System.*out*.println("sign=" + sign);

                treeMap.put("sign",sign);

//拼接微信统一下单参数,xml格式

                keys = treeMap.keySet();

                StringBuffer wxPostXML = new StringBuffer();

                wxPostXML.append("<xml>");

                for (String key : keys){

                    String value = treeMap.get(key);

                    wxPostXML.append("<"+key+">"+value+"</"+key+">");

                }

                wxPostXML.append("</xml>");

                System.*out*.println("wxPostXML=" + wxPostXML);

//                请求微信统一下单接口

                String weixinResult = HttpRequestUtils.*sendPost*(Configure.*CREATE\_ORDER\_API*, wxPostXML.toString());

                System.*out*.print(weixinResult);

                Map<String,Object>  orderMap= XMLParser.*getMapFromXML*(weixinResult);

//                判断是否统一下单成功,获取预支付订单信息返回页面,页面用来吊起微信支付

                if("SUCCESS".equals(orderMap.get("return\_code"))){

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

                        resultMap.put("flag",true);

                        resultMap.put("orderId",orderVo.getId());

                        resultMap.put("appId",Configure.*getAppid*());

                        resultMap.put("timeStamp",new Date().getTime()/1000);

                        resultMap.put("nonceStr", RandomStringGenerator.*getRandomStringByLength*(32));

                        resultMap.put("signType","MD5");

                        resultMap.put("package","prepay\_id="+orderMap.get("prepay\_id").toString());

                        //页面吊起微信支付再次签名==封装有序集合

                        TreeMap<String,String> contentMap = new TreeMap<>();

                        contentMap.put("appId",Configure.*getAppid*());

                        contentMap.put("timeStamp",resultMap.get("timeStamp").toString());

                        contentMap.put("nonceStr", resultMap.get("nonceStr").toString());

                        contentMap.put("package","prepay\_id="+orderMap.get("prepay\_id").toString());

                        contentMap.put("signType","MD5");

                        //页面吊起微信支付再次签名

                        Set<String> pay\_map = contentMap.keySet();

                        StringBuffer pay\_params = new StringBuffer();

                        for (String key : pay\_map){

                            String value = contentMap.get(key);

                            pay\_params.append(key);

                            pay\_params.append("=");

                            pay\_params.append(value);

                            pay\_params.append("&");

                        }

                        pay\_params.append("key="+Configure.*getKeyPay*());

                        resultMap.put("paySign", MD5.*MD5Encode*(pay\_params.toString()).toUpperCase());

                    }else{

                        resultMap.put("flag",false);

                        resultMap.put("msg",orderMap.get("err\_code\_des").toString());

                    }

                }else{

                    resultMap.put("flag",false);

                    resultMap.put("msg",orderMap.get("return\_msg").toString());

                }

            }else{

                resultMap.put("flag",false);

                resultMap.put("msg","支付失败");

            }

        } catch (Exception e) {

            e.printStackTrace();

*log*.debug(e + "->" + e.getMessage());

        }

        return resultMap;

    }

*/\*\**

*\* 支付成功异步通知====增项付款单*

*\*****@param****request*

*\*****@return***

*\*/*

@RequestMapping(value = "callback\_appreciation")

    @ResponseBody

    public String callbackAppreciation(

            HttpServletRequest request) {

        try{

            //从请求对象中获取回调过来的参数

            InputStream inStream = request.getInputStream();

            ByteArrayOutputStream outSteam = new ByteArrayOutputStream();

            byte[] buffer = new byte[1024];

            int len = 0;

            while ((len = inStream.read(buffer)) != -1) {

                outSteam.write(buffer, 0, len);

            }

            outSteam.close();

            inStream.close();

            String resultStr  = new String(outSteam.toByteArray(),"utf-8");

            System.*out*.print(resultStr);

            Map<String,Object>  resultMap= XMLParser.*getMapFromXML*(resultStr);

            String result\_code = resultMap.get("result\_code").toString();//返回状态码

            String return\_code = resultMap.get("return\_code").toString();//SUCCESS/FAIL此字段是通信标识，非交易标识，交易是否成功需要查看result\_code来判断

            //判断返回结果是否成功

            if (("SUCCESS").equals(return\_code) && ("SUCCESS").equals(result\_code)) {

                String is\_subscribe = resultMap.get("is\_subscribe").toString();//用户是否关注公众账号，Y-关注，N-未关注，仅在公众账号类型支付有效

                String out\_trade\_no = resultMap.get("out\_trade\_no").toString();//商户系统的订单号，与请求一致。

                String transaction\_id = resultMap.get("transaction\_id").toString();//微信支付订单号

                String sign = resultMap.get("sign").toString();//签名

                String time\_end = resultMap.get("time\_end").toString();//支付完成时间，格式为yyyyMMddHHmmss，如2009年12月25日9点10分10秒表示为20091225091010。https://pay.[weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4\_2](http://weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4_2)

                String bank\_type = resultMap.get("bank\_type").toString();//银行类型，采用字符串类型的银行标识，银行类型见银行列表https://pay.[weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4\_2](http://weixin.qq.com/wiki/doc/api/jsapi.php?chapter=4_2)

                //重新封装参数变为有序集合否则会影响 微信签名

                TreeMap<String,String> contentMap = new TreeMap<>();

                Set<String> pay\_keys = resultMap.keySet();

                for (String key : pay\_keys){

                    contentMap.put(key,resultMap.get(key).toString());

                }

                //通过回调参数进行签名并对比验证

                StringBuffer pay\_params = new StringBuffer();

                Set<String> pay\_keys\_ok = contentMap.keySet();

                for (String key : pay\_keys\_ok){

                    if (key.equals("sign"))

                        continue;

                    String value = resultMap.get(key).toString();

                    pay\_params.append(key);

                    pay\_params.append("=");

                    pay\_params.append(value);

                    pay\_params.append("&");

                }

                pay\_params.append("key="+Configure.*getKeyPay*());

                System.*out*.print(pay\_params);

                String paySign = MD5.*MD5Encode*(pay\_params.toString()).toUpperCase();

                System.*out*.print(paySign);

                //判断本地签名是否与回调参数中的签名相同==防止非法操作

                if(paySign.equals(resultMap.get("sign"))){

                    AppreciationOrderVo appreciationOrderVo = appreciationOrderService.selectByNo(out\_trade\_no).getResult();

                    //更新订单信息

                    AppreciationOrder order = new AppreciationOrder();

                    order.setId(appreciationOrderVo.getId());

                    order.setOrderStatus(1);

                    order.setPayTime(new Date());

                    order.setSerialNumber(transaction\_id);

                    if(appreciationOrderService.updateByPrimaryKeySelective(order).getResult() > 0){

                        //发送下单成功消息

                        if(order!=null&&appreciationOrderVo.getCustomerId()!=null){

                            CustomerVo custmerVo =customerService.selectByPrimaryKey(appreciationOrderVo.getCustomerId()).getResult();

                            if(custmerVo!=null&&custmerVo.getWecatPlatformId()!=null){

                                WecatSmsTemplateVo wecatSmsTemplateVo = wecatSmsTemplateService.getWecatSmsTemplateByPlatformId(custmerVo.getWecatPlatformId(),2).getResult();

                                WecatTokenVo wecatTokenVo =wecatTokenService.selectWecatTokenByPlatformId(custmerVo.getWecatPlatformId()).getResult();

                                SysSmsVo sysSmsVo =sysSmsService.selectOpenSms().getResult();

                                ServiceNetworkVo serviceNetworkVo =serviceNetworkService.selectByPrimaryKey(appreciationOrderVo.getServiceNetworkId()).getResult();

                                if(wecatSmsTemplateVo!=null&&wecatTokenVo!=null&&sysSmsVo!=null){

//                                    MessageSendUtil.sendMessage(appreciationOrderVo,custmerVo,wecatSmsTemplateVo,null,sysSmsVo,wecatTokenVo,serviceNetworkVo,2);

//                                    需增加增项付款成功的消息通知

                                }

                            }

                        }

                        System.*out*.print("增项付款支付成功后--后台订单同步成功");

*log*.debug("增项付款支付成功后--后台订单同步成功");

                        return "SUCCESS";

                    }else{

                        System.*out*.print("支付成功后--后台订单同步失败");

*log*.debug("支付成功后--后台订单同步失败");

                        return "<xml><return\_code><![CDATA[FAIL]]></return\_code><return\_msg><![CDATA[订单后台同步处理失败!]]></return\_msg></xml>";

                    }

                }else{

                    System.*out*.print("签名失败");

*log*.debug("签名失败");

                    return "<xml><return\_code><![CDATA[FAIL]]></return\_code><return\_msg><![CDATA[签名失败]]></return\_msg></xml>";

                }

            }else{

                System.*out*.print(resultMap.get("result\_msg"));

*log*.debug(resultMap.get("result\_msg")+"");

                return "<xml><return\_code><![CDATA[FAIL]]></return\_code><return\_msg><![CDATA["+resultMap.get("result\_msg")+"]]></return\_msg></xml>";

            }

        }catch (Exception e){

            return "<xml><return\_code><![CDATA[FAIL]]></return\_code><return\_msg><![CDATA[异步通知失败]]></return\_msg></xml>";

        }

    }