微信公众号支付H5调用详解（附代码）

blog.csdn.net/qq_28590639/article/details/49099275

2015年10月13日 15:27:30 阅读数：18085

最近开发的微信公众号需要开启公众号支付功能，经过几天时间的开发，终于搞定了，在这里记录一下开发流程，也给需要的朋友提供一点点帮助。

前面的公众号设置我就不多说了，直接进入代码开发。

1、在微信公众号支付的开发文档中，H5调起支付API页面有说明，需要在微信浏览器的内置对象WeixinJSBridge中调用getBrandWCPayRequest方法，该方法需要的参数为：

公众号id	appId	是	String(16)	wx8888888888888888	商户注册具有支付权限的公众号成功后即可获得
时间戳	timeStamp	是	String(32)	1414561699	当前的时间，其他详见时间戳规则
随机字符串	nonceStr	是	String(32)	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	随机字符串，不长于32位。推荐随机数生成算法
订单详情扩展字符串	package	是	String(128)	prepay_id=123456789	统一下单接口返回的prepay_id参数值，提交格式如：prepay_id=***
签名方式	signType	是	String(32)	MD5	签名算法，暂支持MD5
签名	paySign	是	String(64)	C380BEC2BFD727A4B6845133519F3AD6	签名，详见签名生成算法

因此在调用支付API之前必须准备好上述6个参数。公众号id，时间戳，随机字符串和签名方式都好说，下面介绍如何获取订单详情扩展字符串和签名。

2、获取订单详情扩展字符串

获取订单详情扩展字符串其实就是要获取prepay_id，在微信公众号支付的开发文档的API列表->统一下单页面有获取prepay_id的参数列表，如下为获取prepay_id的方法：

```
/**
* 获取prepay_id
* @return AccessToken
* @author ****
* @date 2015.10.12
*/
@SuppressWarnings("null")
public static String getPrepayId(String userid, String addrip)
```

```
{
//公众账号ID
String appid = Config.APPID;

//商户号
String mch_id = "**********";

//设备号
String device_info = "WEB";

//随机字符串
String nonce_str = String.valueOf(Math.random());

//商品描述
String body = "1yuanmiaosha";

//商品详情
String detail = "miaosha";

//附加数据
String attach = "detail";

//商户订单号
String out_trade_no = "***************************";

//货币类型
String fee_type = "CNY";

//总金额
int total_fee = 1;

//终端IP
String spbill_create_ip = addrip;

//交易起始时间(供测试用)
String time_start = DateUtil.getNowTime().replace("-", "").replace(" ", "").replace(":", "");

//交易结束时间(供测试用)
String time_expire = String.valueOf(Long.valueOf(time_start)+(long)320);

//商品标记(供测试用)
String goods_tag = "****";

//通知地址(供测试用)
String notify_url ="http://www.baidu.com";

//交易类型
String trade_type ="JSAPI";

//商品ID(供测试用)
String product_id = "12235413214070356458058";

//指定支付方式
String limit_pay = "no_credit";
```

```
//用户标识
String openid = userid;

//签名(最后获取)
String []array = {"appid=".concat(appid), "mch_id=".concat(mch_id),
"device_info=".concat(device_info),
"nonce_str=".concat(nonce_str), "body=".concat(body), "detail=".concat(detail),
"attach=".concat(attach), "out_trade_no=".concat(out_trade_no), "fee_type=".concat(fee_type),
"total_fee=".concat(String.valueOf(total_fee)),
"spbill_create_ip=".concat(spbill_create_ip), "time_start=".concat(time_start),
"time_expire=".concat(time_expire),
"goods_tag=".concat(goods_tag), "notify_url=".concat(notify_url),
"trade_type=".concat(trade_type),
"product_id=".concat(product_id), "limit_pay=".concat(limit_pay), "openid=".concat(openid)};

//对数组进行字典排序
SignUtil.sort(array);

//组合成stringA
String stringA = new String() ;

String stringSignTemp = new String();

String sign = new String();

String xmlStr = new String();

String prepay_id = new String();

String key = "********************************";

for(int i = 0 ; i < array.length ; i++)
{
stringA = stringA.concat(array[i].concat("&"));
}

stringSignTemp = stringA.concat("key=").concat(key);

//对stringSignTemp进行MD5加密
MySecurity mySecurity = new MySecurity();

sign = mySecurity.encode(stringSignTemp, "MD5");

//将sign所有的字母换成大写  得到sign
sign = sign.toUpperCase();

System.out.println("sign 为"+sign);

Map<String, String> paraMap = new HashMap<String, String>();

paraMap.put("appid", appid);
paraMap.put("attach", attach);
paraMap.put("body", body);
paraMap.put("detail", detail);
paraMap.put("device_info", device_info);
paraMap.put("fee_type", fee_type);
paraMap.put("goods_tag", goods_tag);
paraMap.put("limit_pay", limit_pay);
paraMap.put("mch_id", mch_id);
paraMap.put("nonce_str", nonce_str);
paraMap.put("notify_url", notify_url);
paraMap.put("openid", openid);
paraMap.put("out_trade_no", out_trade_no);
paraMap.put("product_id", product_id);
paraMap.put("spbill_create_ip", spbill_create_ip);
paraMap.put("time_expire", time_expire);
paraMap.put("time_start", time_start);
paraMap.put("total_fee", String.valueOf(total_fee));
paraMap.put("trade_type", trade_type);
paraMap.put("sign", sign);

String requestUrl = "https://api.mch.weixin.qq.com/pay/unifiedorder";
String xml = ArrayToXml(paraMap);
//System.out.println(" xml 为"+ xml);
xmlStr = HttpUtil.sendHttpsPOST(requestUrl, xml);
//转换编码格式
String start = "<prepay_id><![CDATA[";
String end = "]]></prepay_id>";
prepay_id = xmlStr.substring(xmlStr.indexOf(start)+start.length(), xmlStr.indexOf(end));
return prepay_id;
}

 //用于字典排序
    public static void sort(String a[]) {
        for (int i = 0; i < a.length - 1; i++) {
            for (int j = i + 1; j < a.length; j++) {
                if (a[j].compareTo(a[i]) < 0) {
                    String temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
    }

public String encode(String strSrc, String encodeType) {
MessageDigest md = null;
String strDes = null;
byte[] bt = strSrc.getBytes();
try {
if (encodeType == null || "".equals(encodeType))
encodeType = MD5;//默认使用MD5
md = MessageDigest.getInstance(encodeType);
md.update(bt);
```

```
strDes = bytes2Hex(md.digest());
} catch (NoSuchAlgorithmException e) {
return strSrc;
}
return strDes;
}



/**
* map转成xml
*
* @param arr
* @return
*/
public static String ArrayToXml(Map<String, String> arr) {
String xml = "<xml>";

Iterator<Entry<String, String>> iter = arr.entrySet().iterator();
while (iter.hasNext()) {
Entry<String, String> entry = iter.next();
String key = entry.getKey();
String val = entry.getValue();
xml += "<" + key + ">" + val + "</" + key + ">";
}

xml += "</xml>";
return xml;
}



public static String sendHttpsPOST(String url, String data) {
String result = null;

try {
// 设置SSLContext
SSLContext sslcontext = SSLContext.getInstance("TLS");
sslcontext.init(null, new TrustManager[] { myX509TrustManager },
null);

// 打开连接
// 要发送的POST请求url?Key=Value&amp;Key2=Value2&amp;Key3=Value3的形式
URL requestUrl = new URL(url);
HttpsURLConnection httpsConn = (HttpsURLConnection) requestUrl
.openConnection();

// 设置套接工厂
httpsConn.setSSLSocketFactory(sslcontext.getSocketFactory());
```

```
// 加入数据
httpsConn.setRequestMethod("POST");
httpsConn.setDoOutput(true);
OutputStream out = httpsConn.getOutputStream() ;

if (data != null)
out.write(data.getBytes("UTF-8"));
out.flush();
out.close();

// 获取输入流
BufferedReader in = new BufferedReader(new InputStreamReader(
httpsConn.getInputStream()));
int code = httpsConn.getResponseCode();
if (HttpsURLConnection.HTTP_OK == code) {
String temp = in.readLine();
/* 连接成一个字符串 */
while (temp != null) {
if (result != null)
result += temp;
else
result = temp;
temp = in.readLine();
}
}
} catch (KeyManagementException e) {
e.printStackTrace();
} catch (NoSuchAlgorithmException e) {
e.printStackTrace();
} catch (MalformedURLException e) {
e.printStackTrace();
} catch (ProtocolException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}

return result;
}
```

至此已经获取到了prepay_id参数。

3、获取appId、timeStamp、nonceStr、package、signType、paySign参数传至JS页面

```
/**
* 获取appId、timeStamp、nonceStr、package、signType、paySign
* @param userinfo
* @return string
```

```
*/
public String getParameter()
{

BrandWCPayParameter brandWCPayParameter = new BrandWCPayParameter();

String key = "********************************";

//从页面将用户openid传入
String userid = this.getRequest().getParameter("userid");

String addrip = this.getRequest().getParameter("addrip");

//appId
String appId = Config.APPID;

//timeStamp
String timeStamp = String.valueOf(System.currentTimeMillis()/1000);

//nonceStr随机字符串
String nonceStr=String.valueOf(Math.random());

String signType = "MD5";

//package订单详情扩展字符串  prepay_id
String prepay_id = new String();

prepay_id = HttpUtil.getPrepayId(userid,addrip);

String packAge = "prepay_id=" + prepay_id;

//获取签名
String []array = {"appId=".concat(appId), "timeStamp=".concat(timeStamp),
"nonceStr=".concat(nonceStr),
"package=".concat(packAge), "signType=".concat(signType)};

SignUtil.sort(array);

//组合成stringA
String stringA = new String() ;

String stringSignTemp = new String();

String paySign = new String();

for(int i = 0 ; i < array.length ; i++)
{
stringA = stringA.concat(array[i].concat("&"));
}

stringSignTemp = stringA.concat("key=").concat(key);
MySecurity mySecurity = new MySecurity();
paySign = mySecurity.encode(stringSignTemp, "MD5");
//将sign所有的字母换成大写 得到sign
paySign = paySign.toUpperCase();

brandWCPayParameter.setAppId(appId);
brandWCPayParameter.setNonceStr(nonceStr);
brandWCPayParameter.setPackAge(packAge);
brandWCPayParameter.setPaySign(paySign);
brandWCPayParameter.setSignType(signType);
brandWCPayParameter.setTimeStamp(timeStamp);

List<BrandWCPayParameter> list = new ArrayList<BrandWCPayParameter>();
list.add(brandWCPayParameter);

JsonConfig jsonConfig = JsonUtil.configJson("yyyy-MM-dd HH:mm:ss");
JSONArray json = JSONArray.fromObject(list, jsonConfig);
this.showjsondata(json.toString());
return null;
}
```

4、页面调用H5支付API

```
function onBridgeReady(){
//获取appId、timeStamp、nonceStr、package、signType、paySign
var addrip = "<%=addrip%>";
var userid = "<%=user%>";
var appId;
var timeStamp;
var nonceStr;
var signType;
var packAge;
var paySign;
$.post('/weixin/cj/Cjaction!getParameter.action',{addrip:addrip,userid:userid},function(data)
{
var brandWCPayParameter =data[0];
appId = brandWCPayParameter.appId;
timeStamp = brandWCPayParameter.timeStamp;
nonceStr = brandWCPayParameter.nonceStr;
signType = brandWCPayParameter.signType;
packAge = brandWCPayParameter.packAge;
paySign = brandWCPayParameter.paySign;
alert(paySign);
WeixinJSBridge.invoke(
'getBrandWCPayRequest', {
"appId" : appId,     //公众号名称，由商户传入
"timeStamp": timeStamp,         //时间戳，自1970年以来的秒数
"nonceStr" : nonceStr, //随机串
"package" : packAge,
"signType" : signType,         //微信签名方式:
"paySign" : paySign    //微信签名
},

function(res){
if(res.err_msg == "get_brand_wcpay_request:ok" ) {
  alert("支付成功");
}
}
);
},"json");
}



//微信支付方法（点击按键调用）
function pay(){
if (typeof WeixinJSBridge == "undefined"){
 if( document.addEventListener ){
     document.addEventListener('WeixinJSBridgeReady', onBridgeReady, false);
 }else if (document.attachEvent){
     document.attachEvent('WeixinJSBridgeReady', onBridgeReady);
     document.attachEvent('onWeixinJSBridgeReady', onBridgeReady);
 }
}else{
 onBridgeReady();
}
}
```

5、注意事项

获取prepay_id参数时，当商品描述、商品详情、附加数据三个参数为中文时会返回乱码或者签名错误，应该是编码格式的原因，但是具体是什么原因还不是很清楚，如有遇到相同情况的朋友且已解决还请告知；

一定要注意参数的大小写，本人由于大小写原因导致签名失败而浪费了很多时间；

由于急于调通支付功能，代码中存在一些不合理的地方还待优化，请自行甄别；

上述代码中部分方法定义于不同的类中，请自行组织不同类的方法。

6，如果上述材料能对您有任何帮助那么我倍感欣慰，下面是我开发的公众号二维码，有兴趣的朋友可以关注一下！！谢谢！！

阅读更多

个人分类： 微信公众号开发

微信<em>支付</em>开发(3) JS API<em>支付</em>

本文介绍如何使用JS API<em>支付</em>接口完成微信<em>支付</em>。 一、JS API<em>支付</em>接口（getBrandWCPayRequest） 微信JS API只能在微信内置浏览器中使用，其他浏览器<em>调用</em>无效。微信...

没有更多推荐了，返回首页

个人资料

原创

1

粉丝

0

喜欢

0

评论

1

等级：

访问：
1万+

积分：
106

排名：
127万+

个人分类

微信公众号开发	1篇

归档

热门文章

微信公众号支付H5调用详解（附代码）
阅读量：18076