**支付接入文档 V1.3**

一、 下单接口  
该接口做初始化支付的相关请求，通过该接口得到支付信息。建议通过服务器进行交互  
请求方式:POST  
请求格式:JSON  
请求接口： http://www.kanuoyishu.com:9091/business/order/prepareOrder   
**注意请求头添加: Content-Type: application/json  
注意：所有的支付类型得到的返回结果都必须先访问。无需做任何处理。访问后会得到一个具体的可支付或者支付的结果。切记不要对地址进行任何封装**

**如果不加会出现“参数异常”错误**请求参数列表如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **变量名** | **类型** | **说明** |
| 版本号 | version | String | 默认 1.0 |
| 商户编号 | merId | String | 商户后台获取 |
| 商户订单号 | orderId | String(32) | 商户唯一订单号 |
| 订单金额 | totalMoney | String | 单位:分， 必须大于 100 分 |
| 支付类型 | tradeType | String | 支付宝： alipay 微信： wechat 银联: unionpay QQ 钱包: qqpay  京东支付:jdpay 苏宁支付:snpay |
| 请求ip | ip | String | 请求IP， 部分支付请求IP与支付IP必须一致，所以请上传用户请求的真实IP |
| 商品描述 | describe | String | 商品描述 |

|  |  |  |  |
| --- | --- | --- | --- |
| 异步通知 URL | notify | String | 数据异步通知(可在后台配置，后台配 置的通知地址优先) |
| 同步跳转URL | redirectUrl | String | 不能带有任何参数(某些通道无效) |
| 订单备注说明 | remark | String(64) | 可为空， 如果传递必须为字符串或者 数据组合 |
| 支付来源 | fromtype | String | wap : 普通 wap |
| md5签名串 | sign | String | 参照签名校验规则 |

签名校验规则如下：  
{value}要替换成接收到的值， {key}要替换成平台分配的接入密钥，可在商户后台获取.  
merId={value}&orderId={value}&totalMoney={value}&tradeType={value}&{key}

注意:括号{}要去掉,

如:

merId=10003&orderId=123456789987&totalMoney=100&tradeType=alipay&ac8d6f81091e49cfb4769de549dbb770  
使用 md5 签名上面拼接的字符串即可生成 32 位密文再转换成大写

请求示例：  
{"version":"1.0","merId":"10001","notify":"","orderId":"T1491445348661","redirectUrl":"","remark":"hello","sign":"503374D0333610080A03F602C5A3994E","totalMoney":"1","tradeType":"alipay ","describe":"测试商品"， " fromtype ":" wap"}  
返回结果：  
结果按 json 方式实时返回

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **变量名** | **类型** | **说明** |
| 状态码 | code | int | 0 成功 非 0 失败 |
| object | tradeType | String | 支付类型 |
| token | String | 暂保留 |
| wxPayinfo | String | 暂保留 |
| wxAppId | String | 暂保留 |
| wxPayWay | String | 微信地址(老版本保留) |
| aliPayWay | String | 支付宝地址(老版本保留) |
| data | String | 支付地址(存在则使用该值进行支付) |

返回示例:  
{"code":0,"errMsg":"","object":{"tradeType":"pay.alipay.native","token":"","wxPayinfo":"","wxAppId":"","wxPayWay":"","aliPayWay":"https://qr.alipay.com/bax002640xkv1hfkwfd52078","data":"https://qr.alipay.com/bax002640xkv1hfkwfd52078"}}

二、 订单查询接口  
商户可以主动发起订单查询。  
请求方式:POST  
请求格式:JSON  
请求接口： http://www.kanuoyishu.com:9091/business/order/query  
**注意 HTTP 请求头添加: Content-Type: application/json**  
请求参数:

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **变量名** | **类型** | **说明** |
| 商户编号 | merId | String | 商户后台获取 |
| 商户订单号 | orderId | String | 商户唯一订单号 |
| md5 验证签名串 | sign | String | 参照签名规则 |

签名规则：  
{value}要替换成接收到的值， {key}要替换成平台分配的接入密钥，可在商户后台获取merId={value}&orderId={value}&{key}  
注意:{} 需要去掉  
使用 md5 签名上面拼接的字符串即可生成 32 位密文再转换成大写  
请求示例 ：  
{"merId":"10001","orderId":"T1491445348661","sign":"503374D0333610080A03F60 2C5A3994E"}  
返回结果：  
返回信息同步返回，以 json 方式提供。

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **变量名** | **类型** | **说明** |
| 状态码 | status | String | 返回状态，0 为成功，非 0 为订单未付款或失败 |
| 返回信息 | msg | String | 具体信息描述 |
| 订单号 | orderId | String | 订单号 |
| 金额 | money | String | 金额 |

返回示例：  
{"status":"0","msg":""," orderId ":"商户订单号"," money ":"100"}

三、 计费通知接口  
商户下单传递的 notify 或者在后台配置的通知地址，通过以下规则通知到商户接口  
请求方式:GET  
请求接口:由商户提供(优先使用后台配置地址,如接口和后台同时配置地址，优先使用后台配置的地址)  
请求参数列表

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **变量名** | **类型** | **说明** |
| 状态 | code | String | 0 成功 |
| 商户号 | merId | String | 后台分配的商户号 |
| 商户订单号 | orderId | String | 商户下单订单号 |
| 平台订单号 | tradeId | String | 平台交易订单号 |
| 支付类型 | payWay | String | 1:微信  2:支付宝  3:银联  4:QQ 钱包  5:京东  6:苏宁 |
| 价格 | money | String | 分，支付金额 |
| 自定义参数 | remark | String | 下单透传自定义参数 |
| 计费时间 | time | String | yyyymmddhhmmss |
| 签名串 | sign | String | 参照签名说明 |

通知签名规则如下:  
使用接收到的值替换每个参数的{value},{key}使用商户 key  
code{value}merId{value}money{value}orderId{value}payWay{value}remark{value}time{value}tradeId{value}{key}  
然后 md5 签名上面拼接的字符串即可生成 32 位密文并转为大写  
注意:{} 需要去掉  
请求示例：  
<http://www.baidu.com/?code=0&merId=10003&money=1&orderId=T1493954846592&payWay=1&remark=hello&sign=97D0FBD267B90A91B03F392AD9B2638F&time=20170505112728&tradeId=T20170505112728965199>

返回结果： success  
只同步成功数据,接收到数据后请返回 success,其他返回均当作失败处理,间隔 5 分钟后再次同步，最多同步 5次。