**接口对接规范**

**2018年10月30日**

# 对接说明

域名=jings.wang

例：访问加密请求接口时：

<http://www.jings.wang/api/pay/getEncryption>

备注：

1：接口所有字段的字母均小写。

2：所有请求都是POST.其余请求模式都会被拒绝

3：回调不一定稳定，建议都加上主动查询接口

## 加密接口

|  |  |  |
| --- | --- | --- |
| **接口说明** | 加密请求 | |
| **发起方** | 客户平台 | |
| **接收方** | 支付平台 | |
| **接口类型** | http请求（post） | |
| **请求地址** | [/api/pay/getEncryption](http://localhost:3000/swagger-ui.html" \l "/operations/pay-api-controller/getEncryptionUsingPOST) | |
| **参数字段** | **字段说明** | **字段要求** |
| | appid | | --- | | appid | 必填 |
| params | 需要加密的字符串 |  |
|  |  |  |
| **返回值：模板**  **{**  **"msg": "OK",**  **"status": "OK",**  **"data": "649B291CE79EB6FA963BEDD66946FDFB"**  **}** | | |
| **加密字符串模板：orderid=1&ordername=订单名称** | | |
| **只要符合以上格式都可以，可以加任意多参数进行混淆**  **注：除本加密接口，其他所有接口都需要经过本接口加密后，以密文来访问** | | |

## 1.2支付回调

用户付款完成后，支付平台会向支付接口传入的回调通知接口【notify\_url】发送回调请求。客户业务处理成功，返回字符串success ，即不会继续回调，否则会继续回调，总共回调10次，间隔1分钟

|  |  |  |
| --- | --- | --- |
| **接口说明** | 支付结果回调 | |
| **发起方** | 支付平台 | |
| **接收方** | 客户平台 | |
| **接口类型** | http请求,接收方请不要设置请求模式限制 | |
| **请求地址** | 支付请求的【notifyurl】字段 | |
| **参数字段** | **字段说明** | **字段要求** |
| payid | 支付id（支付平台生成的id，对账时可用） | 必填 |
| orderid | 客户平台在发起付款接口传入的自定义订单号 | 必填 |
| orderuid | 客户平台在发起付款接口带入的自定义参数 | 必填 |
| paymoney | 客户平台在发起付款接口传入的支付金额 | 必填 |
| realpay | 实际支付的金额 | 必填 |
| 回调请求参数模板 | {"orderNo":"be8b505854ba4fb88be0f215a17da78d","orderuid":"12","paymoney":"0.01","realprice":"0.01","payStatus":"1","createTime":"2018-10-18 19:41:38"} | 回参 |
| orderNo:平台单号（全平台唯一）  createTime:平台订单创建时间，请记录好，对账以此时间为标准做时间点切割 payStatus:支付状态 0：待支付 1：支付成功 其他值：支付失败 | | |

## 1.3支付（下单接口）

|  |  |  |
| --- | --- | --- |
| **接口说明** | 下单（支付） | |
| **发起方** | 客户 | |
| **接收方** | 平台 | |
| **接口类型** | http POST请求 | |
| **请求地址** | /api/pay/getPayJson | |
| **参数字段** | **字段说明** | **字段要求** |
| params | 支付参数 | 必填 |
| appid | appid | 必填 |
| 请求流程：  1：将拼接的参数拿到加密接口加密  2：将加密后的密文放到params参数进行请求 | | |
| 返回值模板：  {  "msg": "OK",  "status": "OK",  "data": {  "orderid": "321",  "orderNo": "e9ecb896db054930b1a203958b09a68c",  "orderuid": "12",  "paymoney": "0.01",  "createTime": "2018-10-19 10:23:08",  "payCode": "https://qr.alipay.com/bax07777gxvytldzlthx4083",  "type": "JSON"  }  }  返回值说明：  orderNo：我们的平台订单编号 全平台唯一，查支付状态也用这个来查  createTime：创建时间，对账时以此时间为基准  payCode：支付码，（1）可以拿去封装做成H5跳转模式（2）直接通过工具生成二维码放到网站上让客户使用支付宝扫，一个码只能用一次  orderNo:平台单号  type:(1)JSON 指定返回的是一个支付码；（2）FORM 指定返回的是一个FORM表单源码，直接嵌入webView或者跳转到浏览器，即可自动唤醒支付  3：URL 【场景1：调起客户手机浏览器，比如UC或者QQ浏览器，打开payCode字段的链接，通过浏览器跳转成功率高】【场景2：使用app的webView打开payCode字段的链接，IOS兼容性差劲，容易不能唤醒】  IOS的WEBView跳转解决方案：拦截我们的阿里跳转协议 | | |
| 完整请求参数拼接案例：  orderid=SH-98d6cffa-d742-4f30-9637-0bffb76b0b4b&ordername=订单名称&paymoney=63&orderuid=客户:514&paytype=11&notifyurl=http://baidu.com&returnurl=https://baidu.com&orderinfo=订单说明&appid=418a21be35164903a18ef2485c886ac7 =========================字段说明============================  orderid：商户订单号，必须唯一，否则下单失败  orderuid:下单客户id，必须填  paymoney：支付金额  paytype：支付宝（11） 暂时只有支付宝  notifyurl:回调地址（支付成功后平台对此地址进行通知）  returnurl：支付成功前台跳转地址 | | |

## 1.4订单查询接口

商户平台获取客户订单支付状态，在正常情况可以通过支付通知接口【notify】获取数据即可；在异常情况，例如服务器中断或网络中断等原因 ，可用该接口查询订单状态。注意：此接口不能频繁查询,一个订单号一分钟只能查询一次。

|  |  |  |
| --- | --- | --- |
| **接口说明** | 订单状态查询 | |
| **发起方** | 商户平台 | |
| **接收方** | 支付平台 | |
| **接口类型** | http请求（POST） | |
| **请求地址** | [/api/pay/queryPayStatus.do](http://localhost:3000/swagger-ui.html" \l "/operations/pay-api-controller/queryPayStatusUsingPOST) | |
| **参数字段** | **字段描述** | **字段说明** |
| orderNo | 订单No | 入参 |
| appid | appid | 入参 |
| 回参报文 | 回参为json数据，具体如下：  {  "msg": "OK",  "status": "OK",  "data": {  "orderNo": "be8b505854ba4fb88be0f215a17da78d",  "orderuid": "12",  "payStatus": "1",  "createTime": "2018-10-18 19:41:38",  "paymoney": "0.01",  "realprice": "0.01"  }  } | 回参 |
| 与回调返回参数一样  payStatus:支付状态 0：待支付 1：支付成功 其他值：支付失败 | | |
| 请求案例：  1：将 “orderNo=1”拿到加密接口加密  2：拿到加密后的参数，请求此接口 | | |