

adb支付平台接口文档

准备工作

- 获取接口访问密钥, 假设密钥key为 4f6168398ae711eb24f72fb86638796f
- 获取接口网关地址, 假设接口网关为 http://192.168.1.1:8008/api/adb_paygate/new

接口说明

- 提交的HTTP方法为**POST**
- 接口通信中有字段是中文的, 请务必使用 **utf-8** 编码
- 接口提交和响应数据格式均为 **JSON**字符串
- 提交Header: Content-Type: application/json
- 提交Body样例:

```
{
    "out_order_id": "000082",
    "qrtype": "alipay",
    "fee": 19,
    "notify_url": "http://requestbin.net/ve1",
    "attach": "anything",
    "sign": "CA34B9B7CDFA4A9ACFE44A56939E8A79"
}
```

字段说明:

out_order_id:	必填: 外部系统订单ID, 必须保证唯一, 长度1-64
qrtype:	必填: 支付类型, 目前可选: alipay
fee:	必填: 金额, 单位RMB分, 范围1-10000000000
notify_url:	可选: 接收回调的地址, 必须是http或https, 最大长度128
attach:	可选: 任意自定义信息, 回调的时候会原样返回, 最大长度128
sign:	必填: 签名, 详见下面的签名算法, 长度1-64

- 响应Body样例:

```
{
    "code": 1,                                // 1表示成功, 其他表示失败
    "message": "",                            // code不为1时的错误信息
    "qrimage": "data:image/png;base64...", // 二维码, 支付宝扫码转账
    "order_id": "2019620183258-BA01",        // 平台订单ID
    "out_order_id": "000083",                  // 外部系统订单ID, 原样返回
    "fee": 19,                                // 订单金额(单位分)原样返回
    "fee_yuan": 0.19,                          // 订单金额(单位元)
    "time": "2019-06-20T18:32:58.669732602+08:00"
}
```

回调说明

- 如果支付请求时提交的**notify_url**不为空, 则当订单支付成功后, 会向该**notify_url**地址发送异步回调通知
- 回调推送提交的HTTP方法为**POST**
- 接收通知的服务器请在5秒钟内响应, http状态码200则标记通知成功, 其它状态码则标记通知失败, 回调推送结束

- 为保障推送到达率，系统可能多次进行通知推送，请做好去重逻辑
- 失败的异步回调会自动进行**重复推送**，重试间隔为(10s, 30s, 120s, 300s, 900s)
- 为了防止伪造的回调，接收通知的系统对于回调通知的字段一定要**验签**，详见下面的签名算法
- 回调通知格式：

```
{  
    "code": 1,                                // 1表示成功，其他表示失败  
    "out_order_id": "000082",                  // 外部系统订单ID，原样返回  
    "fee": 19,                                 // 订单金额(单位分)原样返回  
    "attach": "anything",                     // 提交订单时的自定义信息，原样返回  
    "sign": "CA34B9B7CDFA4A9ACFE44A56939E8A79", // 签名，需接收端校验  
    "time": "2019-06-20T16:18:53.616803582+08:00"  
}
```

签名算法

按如下步骤生成签名

- **请求支付和回调通知中均包含字段out_order_id fee**，分别代表外部系统订单ID 订单金额
- 拼接字符串out_order_id={out_order_id}&fee={fee}&key={接口密钥}
- 对拼接所得的字符串进行MD5加密
- 将加密所得字符串全部转换为大写

最后得到的就是签名值