

云支付 开发文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

开发文档

HTTPS API 说明

接口调用说明

枚举值定义

公共数据结构

交易接口

刷卡支付

扫码支付

公众号支付

H5 支付

APP 支付

申请退款

查询订单

关闭订单

查询退款单

支付成功回调

刷卡押金冻结

扫码押金冻结

押金消费

押金解冻

查询押金冻结单

查询押金解冻单

多条件查询订单信息

子商户交易汇总查询

门店接口

查询门店信息

支付场景

刷卡支付场景

扫码支付场景

SDK 说明

SDK 简介

SDK 接入流程

机具接入说明

机具对接流程

收款机具 API 接入流程

开发文档

HTTPS API 说明

接口调用说明

最近更新时间：2020-06-03 11:42:00

通讯说明

- 所有接口均使用 HTTPS 通信，数据包格式为 json（HTTP 请求的 content-type 字段必须使用 application/json）。
- 请求必须传认证信息，需要传认证码和认证算法。
- 响应要验证认证码。
- 所有接口参数名使用的字母均为小写。

数据包格式说明

请求包格式

请求格式示例（以 query_order 接口为例）：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "69E65DA0A32F18D059DBE81CA4D9D702470184966E12A03715584249788BD8DD"
    }
  },
  "request_content": "{
    \"pay_mch_key\": {\"pay_platform\": 1, \"sub_pay_platform\": 100, \"out_shop_id\": \"sz011biKxOguirmBqiFR\", \"out_s
ub_mch_id\": \"sz01KzuCUOmw8yjtPite\", \"out_mch_id\": \"sz01IXKA6DKGjNzr2l4B\"}, \"trade_type\": 1, \"out_trade_no\
\": \"sz010002cz11564386781\", \"nonce_str\": \"E94C00688C3F429CA2B0B396BF823548\", \"order_client\": {\"staff_id\
\": \"1192\", \"machine_no\": \"1111\", \"terminal_type\": 1, \"sdk_version\": \"1.0\", \"device_id\": \"12345\", \"spbill_crea
te_ip\": \"90.0.0.0\"}}}"
}
```

- 详解：**
- 请求包含两个字段：authen_info 和 request_content。前者表示认证信息，为 json 结构；后者表示请求具体内容，为 json 形式字符串化。
- request_content 为具体请求内容的 json 字符串化结构，见各具体接口。
 - authen_info 结构有嵌套属性 a，字段为认证码：

字段	类型	说明
authen_type	Number(32)	认证算法类型。只支持填1，即为 HMAC-SHA256
authen_code	String(64)	认证码

认证码生成算法：HMAC-SHA256 认证密钥为服务商在云支付录入商户时，在子商户页面上生成的认证密钥。

响应包格式

响应包与请求包类似，包含两个字段：authen_info 和 response_content。前者表示认证信息（响应包仅有认证码方式），为 json 结构；后者表示响应具体内容，为 json 形式字符串化。响应格式示例（以交接班接口为例）：

```
{
  "authen_info": {
    "a": {
```



```
"authen_type": 1,
"authen_code": "1CB622818DF1E2D91741A5FE792F1EFDD2557343FBD1275628E832A95CBB0FBC"
},
"response_content": "
{"status":0,"description":"\\u64CD\\u4F5C\\u6210\\u529F\\u3002","log_id":1167366844,"internal_status":0}"
}
```

其它说明

各种 update 接口中，如选填字段不填写，则该字段不需修改。清空此字段时，需上传此字段的内容为空。

● 接口调用说明

交易接口中的门店信息，必须和子商户在云支付手机端商户管理系统设置的一致。

● 订单和退款单号说明

为了保护不同商户的订单号不重复，云支付为每个服务商录入的子商户分配“云支付订单前缀”，在云支付后台的商户详情中可以看到，该商户的订单和退款单必须以云支付子商户号做前缀。

数据包构造示例

下面以刷卡支付为例来介绍整个认证过程：

客户端向服务器发送请求

1. 判定使用场景。

在这个示例中，我们要发出的是刷卡支付请求。因此，需要使用认证算法来计算认证码。

2. 根据接口的输入参数，构造 json 格式字符串。

```
"request_content": {
  "pay_mch_key": {
    "pay_platform": 2,
    "out_mch_id": "1234mcWYS3M5TjKLorAZ",
    "out_sub_mch_id": "12343ycHpBDv8GXJfmSv",
    "out_shop_id": "1234ruQC1eTa9w30AaAH",
  },
  "nonce_str": "542AB309ECA042FE92355BDEC4E2D733",
  "order_client": {
    "staff_id": "shop_manage_id_01",
    "machine_no": "34-64-a9-15-b4-c1",
    "terminal_type": 1,
    "sdk_version": "1.6",
    "spbill_create_ip": "10.27.14.138",
    "device_id": "device_id_01"
  },
  "pay_content": {
    "author_code": "282129340414399818",
    "out_trade_no": "12341008b320170802191960015",
    "body": "生活用品套餐",
    "total_fee": 1,
    "fee_type": "CNY"
  }
}
```

3. 将 json 格式字符串转换为字符串，并使用认证算法，认证 key 来计算认证码。

```
A2F2F3C3506F3461212525C4917479B515ABB42BDC5909F7C012B6F74C0C1B99
```

4. 构造 authen_info 结构。

```
"authen_info":
{
  "a":{
    "authen_type":1,
    "authen_code":"A2F2F3C3506F3461212525C4917479B515ABB42BDC5909F7C012B6F74C0C1B99"
  }
}
```

5. 构造 json 格式的请求数据包，需要包含 authen_info 和 request_content 字段。

6. 将 json 格式请求转换为字符串，发送给服务器。

```
"request_content":{"pay_mch_key":{"pay_platform":2,"out_mch_id":"1234mcWYS3M5TjKLorAZ",
"out_sub_mch_id":"12343ycHpBDv8GXJfmSv","out_shop_id":"1234ruQC1eTa9w30AaAH"},"pay_content":{"out_trade_no":"12341008b320170802191960015","author_code":"282129340414399818","total_fee":1,"fee_type":"CNY","body":"生活用品套餐","wxpay_pay_content_ext":{"attach":"","goods_tag":""}},"order_client":{"device_id":"device_id_01","staff_id":"shop_manage_id_01","terminal_type":1,"machine_no":"34-64-a9-15-b4-c1","sdk_version":"1.6","spbill_create_ip":"10.27.14.138"},"nonce_str":"542AB309ECA042FE92355BDEC4E2D733"},"authen_info":{"a":{"authen_type":1,"authen_code":"A2F2F3C3506F3461212525C4917479B515ABB42BDC5909F7C012B6F74C0C1B99"}}},
```

客户端验证服务器应答消息

1. 将响应包从 string 转换为 json 格式消息。
2. 取出 json 格式消息中的 authen_info 和 response_content 字段信息。
3. 使用认证算法和认证 key 对 response_content 计算认证码。
4. 将计算得到的认证码和 authen_info 中的 authen_code 进行比较；正确情况下，两者应该一致。

代码示例 (C++)

构造请求字符串（以刷卡支付为例）：

```
std::string MicroPay_request_str()
{
    /* 1.构造业务请求参数 */
    Json::Value pay_mch_key;    // 构造 pay_mch_key
    pay_mch_key["pay_platform"] = 1;
    pay_mch_key["out_mch_id"]   = "sz013NzuonO6CMJd0rCB";
    pay_mch_key["out_sub_mch_id"] = "sz01ELTR281OFpmdAp6j";
    pay_mch_key["out_shop_id"]   = "sz01qyoPJmd3j1hWmul4";
    Json::Value pay_content;    // 构造 pay_content
    pay_content["out_trade_no"]  = "sz0100lmmnx20171228151031";
    pay_content["author_code"]   = "134680423163089456";
    pay_content["total_fee"]     = 1;
    pay_content["fee_type"]      = "CNY";
    pay_content["attach"]        = "attach";
    Json::Value order_client;    // 构造 order_client
    order_client["machine_no"]   = "32-62-A8-14-B3-C0";
    order_client["sdk_version"]  = "1.0";
    order_client["device_id"]    = 1;
    order_client["spbill_create_ip"] = "192.168.100.75";
    order_client["staff_id"]     = "1003";
    order_client["terminal_type"] = 2;
    Json::Value request_content; // 构造 request_content
    request_content["pay_mch_key"] = pay_mch_key;
    request_content["pay_content"] = pay_content;
    request_content["order_client"] = order_client;
    request_content["nonce_str"]   = "416492026bc84091bcaf7e74ea90ceba";
```

```
Json::FastWriter w;
std::string request_content_str = w.write(request_content);
/* 2. 根据上面的业务请求参数计算认证码 */
Json::Value authen;
std::string hmac;
calc_HMAC_SHA256(authen_key, request_content_str, &hmac); //计算认证码
authen["authen_code"] = hmac;
authen["authen_type"] = 1; //hmac_sha256 为1
Json::Value authen_info;
authen_info["a"] = authen; //认证码
/* 3. 拼装最终请求参数 */
Json::Value request; //构造最终发给服务器的请求
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;
std::string request_str = w.write(request);
return request_str;
}
```

计算认证码：

```
/*
返回是否成功，成功时认证码存放于 hmac 指向的 string
*/
bool calc_HMAC_SHA256(const std::string &key, const std::string &input, std::string *hmac)
{
    unsigned char md[SHA256_DIGEST_LENGTH] = {0}; //32 bytes
    char format_md[65] = {0};
    unsigned int md_len = sizeof(md);
    HMAC_CTX ctx;
    HMAC_CTX_init(&ctx);
    if (!HMAC_Init_ex(&ctx, key.data(), (int)key.length(), EVP_sha256(), NULL) ||
        !HMAC_Update(&ctx, (const unsigned char *)input.data(), input.length()) ||
        !HMAC_Final(&ctx, md, &md_len)) {
        HMAC_CTX_cleanup(&ctx);
        return false;
    }
    HMAC_CTX_cleanup(&ctx);
    for (int i = 0; i < 32; i++) {
        sprintf(&format_md[i * 2], 3, "%02x", md[i]); //二进制转为十六进制大写
    }
    hmac->assign(format_md);
    // 转大写
    transform(hmac->begin(), hmac->end(), hmac->begin(), ::toupper);
    return true;
}
```

枚举值定义

最近更新时间：2023-08-14 17:00:32

交易相关信息

PayPlatform 枚举变量

枚举值	说明
1	微信支付
2	支付宝
4	记账模式

SubPayPlatform 枚举变量

枚举值	说明
100	微信支付
200	支付宝
401	银行卡记账
402	现金记账

TradeType 枚举变量

枚举值	说明
1	刷卡支付
2	扫码支付
3	公众号支付
4	App 支付
5	声波支付
6	手机网站支付
8	一码支付
9	小程序支付

WxpayOrderState 枚举变量

下文中“统一下单”指：公众号支付、一码支付、H5 支付、App 支付、扫码支付等不可直接扣款，需要在微信支付中预生成订单，然后由顾客输入密码的支付方式。

枚举值	说明
1	订单初始态
2	刷卡支付，成功
3	统一下单，支付成功
4	已转入退款

5	刷卡支付，顾客停止支付
6	统一下单，待顾客支付
7	统一下单，订单已关闭
8	刷卡支付，已撤单
9	刷卡支付，用户支付中
10	刷卡支付，支付错误
11	作废状态，表示本地有，第三方支付平台没有的订单
12	订单受理中
13	押金解冻成功
14	押金消费成功

AlipayOrderState 枚举变量

枚举值	说明
1	订单初始态
2	成功
4	等待用户支付
5	已关闭或者已退款
6	交易结束，不可退款
7	订单不存在

RecordOrderState 枚举变量

枚举值	说明
1	订单初始态
2	成功
3	退款

WxpayRefundOrderState 枚举变量

枚举值	说明
1	退款单初始态
2	退款成功
3	退款失败
4	退款处理中
5	转入代发，退款到银行发现用户的卡作废或冻结了，导致原路退款银行卡失败，资金回流到子商户的现金账号，需要子商户人工干预，通过线下或者财付通转账的方式进行退款
6	作废状态，表示本地有，第三方支付平台没有的订单

AlipayRefundOrderState 枚举变量

枚举值	说明
1	退款单初始态
2	退款单成功态
3	申请退款失败
4	退款中
5	订单不存在

RecordRefundOrderState 枚举变量

枚举值	说明
1	退款单初始态
2	退款单成功态

WxpayFreezeOrderState 枚举变量

枚举值	说明
1	冻结单初始态
2	押金刷卡支付，成功
3	等待用户支付
4	押金刷卡支付，顾客停止支付
5	押金冻结金额全部解冻/全部消费
6	押金刷卡支付，支付错误
7	表示本地存在，第三方支付没有订单

AlipayFreezeOrderState 枚举变量

枚举值	说明
1	初始态
2	成功
3	已关闭或者全部已经解冻/消费
4	订单不存在

WxpayUnfreezeOrderState 枚举变量

枚举值	说明
1	解冻单初始态
2	解冻成功
3	解冻失败
4	第三方支付平台没有的解冻单

AlipayUnfreezeOrderState 枚举变量

枚举值	说明
1	解冻单初始态
2	解冻成功
3	解冻失败
4	第三方支付平台没有的解冻单
5	解冻单不存在

认证加密信息

AuthenType 枚举变量

枚举值	说明
1	HMAC-SHA256
2	MD5

接口相关信息

Status 枚举变量

枚举值	说明
0	成功
3	未知
101	根据 description 字段内容，检查调用逻辑是否有问题，如“认证码计算错误”。
102	换新单号重试，并根据 description 字段内容，检查调用逻辑是否有问题，如“单号重复”。
103	隔3秒后原请求重试或查询结果
104	根据 description 字段内容操作，如“退款时顾客余额不足”。 特别提示：在刷卡支付响应包里出现该错误码时，需要判断 internal_status 字段的值是否是407。如果是，则说明客户端发生异常，支付时单号重复，但金额等其他信息不重复，被云支付的防重入挡住，此时，请一定不要撤单，否则会造成已支付的订单退款，给商户造成损失。

公共数据结构

最近更新时间：2023-08-14 17:00:32

订单信息

OrderContent 结构（仅作为返回参数）

一级参数名	二级参数名	必填	类型	说明
out_trade_no	-	是	String(32)	子商户订单号。
transaction_id	-	否	String(32)	第三方支付平台的订单号。
trade_type	-	否	TradeType	交易类型枚举，详见 TradeType。
author_code	-	否	String(128)	刷卡支付时的授权码。
code_url	-	否	String(64)	扫码支付时，用于扫码支付时转换成支付二维码。
time_expire	-	否	Number(64)	订单失效时间（刷卡支付不需要该字段），时间戳（秒）。
time_end	-	否	Number(64)	支付完成时间，时间戳（秒）。
nonce_str	-	是	String(32)	随机字符串。
create_time	-	是	Number(64)	订单创建时间，时间戳（秒）。
last_update_time	-	是	Number(64)	订单最近更新时间，时间戳（秒）。
total_fee	-	否	Number(32)	订单总金额，单位分。
fee_type	-	否	String(3)	货币类型（目前只支持人民币，请填写 CNY）。
cash_fee	-	否	Number(32)	现金支付金额，单位分。
cash_fee_type	-	否	String(3)	现金支付货币类型（目前只支持人民币，请填写 CNY）。
settlement_total_fee	-	否	Number(32)	应结支付金额，单位分。
refunded_fee	-	否	Number(32)	已退款金额，单位分，只有接口 client_order_detail 返回这个字段，接口 query_order 不返回。
body	-	否	String(128)	商品或订单简要描述。
detail	-	否	String(6000)	商品详细列表，由 json 转化而来的 string，json 结构见本节 Detail。
remark	-	否	String(127)	支付备注信息。
coupon_infos	-	否	CouponInfo[]	代金券信息，详见 CouponInfo。
out_freeze_no	-	否	String(32)	此项在押金消费时使用。关联的押金冻结单号。
freeze_fee	-	否	Number(32)	此项在押金消费时使用。押金冻结金额，单位分。
is_confirm_unfreeze	-	否	Bool	此项在押金消费时使用。此项返回是否在消费时解冻用户金额。
manual_unfreeze_fee	-	否	Number(32)	此项在押金消费时使用。消费且解冻金额，单位分。

freeze_id	-	否	String(255)	此项在押金消费时使用。第三方支付授权单号。
wxpay_order_content_ext*	current_trade_state	是	Number(32)	订单当前状态，详见 WxpayOrderState。
	attach	否	String(127)	附加数据，记录子商户自定义数据。
	bank_type	否	String(16)	刷卡支付时特有，付款银行类型，遵守字符型银行编码规范。
	goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
	product_id	否	String(32)	商品 ID，子商户自定义，扫码支付时必须传。
	prepare_id	否	String(64)	公众号或 App 支付时，下单后用于拉起支付的预支付会话标识。
	trade_state_desc	否	String(255)	对当前查询订单状态的描述和下一步操作的指引。
alipay_order_content_ext*	limit_pay	否	String(32)	非刷卡支付时，指定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
	current_trade_state	是	Number(32)	订单当前状态，详见 AlipayOrderState。
	voucher_detail_list	否	AlipayVoucherDetail[]	代金券列表，支付宝回包的内容，详见 AlipayVoucherDetail，示例： <pre>"voucher_detail_list": [{ "id": "20151026000", "name": "XX超市5折优惠", "type": "ALIPAY_FIX_VOUCHER", "amount": 10, "merchant_contribute": 9, "other_contribute": 1, "memo": "学生专用优惠", "purchase_buyer_contribute": 2.01, "purchase_merchant_contribute": 1.03, "purchase_ant_contribute": 0.82 }]</pre>
	fund_bill_list	是	AlipayFundBill[]	支付渠道，支付宝回包的内容，详见 AlipayFundBill，示例： <pre>"fund_bill_list": [{ "fund_channel": "ALIPAYACCOUNT", "amount": 10, "real_amount": 11.21 }]</pre>

	discountable_amount	否	Number(64)	参与优惠的金额。
	undiscountable_amount	否	Number(64)	不参与优惠的金额。
	point_amount	否	Number(64)	积分金额。
	invoice_amount	否	Number(64)	可以开具发票的金额。
	product_code	否	String(32)	产品码。
	royalty_info	否	String(64)	json 的分账信息。
	send_pay_date	否	Number(64)	该订单打款给卖家的时间。
	extend_params	否	String(1024)	扩展信息，花呗相关的逻辑。
	enable_pay_channels	否	String(1024)	可用渠道，多个渠道用 “,” 分割，如： pay_channels="credit_group,point"。
	disable_pay_channels	否	String(1024)	不可用渠道，格式同 enable_pay_channels。
	discount_goods_detail	是	String(1024)	打折相关信息，示例： <pre>[{ "goods_id":"STANDARD1026181538", "goods_name":"雪碧", "discount_amount":"100.00", "voucher_id":"2015102600073002039000002D5O" }]</pre>
	buyer_logon_id	是	String(100)	买家支付宝账号，回包的内容。
	seller_id	否	String(100)	卖家支付宝用户号，回包的内容。
	seller_email	否	String(64)	卖家支付宝账号，回包的内容。
	gmt_refund	是	Number(64)	交易退款时间。
	gmt_close	是	Number(64)	交易结束时间。
	refund_fee	是	Number(64)	退款金额。
	out_biz_no	否	String(64)	商户业务号，回包的内容。
record_order_content_ext*	current_trade_state	是	Number(32)	订单当前状态，详见 RecordOrderState。

带*项只会会有一个。

AlipayFundBill 结构

参数名	必填	类型	说明

fund_channel	是	String(32)	是否发生了资金变化，示例 Y。
amount	是	Number(64)	支付金额。
real_amount	否	Number(64)	实际支付金额。

Detail 结构

参数名	必填	类型	说明
cost_price	否	Number(32)	订单原价，商户侧一张小票订单可能被分多次支付，订单原价用于记录整张小票的支付金额。当订单原价与支付金额不相等则被判定为拆单，无法享受优惠。
receipt_id	否	String(32)	商家小票 ID。
goods_detail	否	GoodsDetail[]	商品详情，详见 GoodsDetail。

GoodsDetail 结构

参数名	必填	类型	说明
goods_id	是	String(32)	商品的编号。
goods_name	否	String(256)	商品名称。支付宝必填，微信支付非必填。
quantity	是	Number(32)	商品数量。
price	是	Number(32)	商品单价（单位：分）。 如果商户有优惠，需传输商户优惠后的单价，单品总金额应 <= 订单总金额 total_fee，否则将无法享受优惠。

AlipayVoucherDetail 结构

参数名	必填	类型	说明
id	是	String(32)	券 ID。
name	是	String(64)	券名称。
type	是	String(32)	代金券类型。
amount	是	Number(64)	代金券金额。
merchant_amount	否	Number(64)	商户出资。
other_amount	否	Number(64)	其它出资。
memo	否	String(256)	备注。

退款单信息

RefundOrderContent 结构（仅作为返回参数）

一级参数名	二级参数名	必填	类型	说明
out_refund_no	-	是	String(32)	子商户退款单号，云支付系统内全局唯一。
refund_id	-	否	String(32)	第三方支付平台的退款单号。
out_trade_no	-	是	String (32)	退款单对应的订单号。
trade_type	-	否	String(128)	TradeType 交易类型，详见 TradeType。

nonce_str	-	是	String(32)	随机字符串。
create_time	-	是	Number(64)	退款单创建时间，时间戳（秒）。
last_update_time	-	是	Number(64)	退款单最近更新时间，时间戳（秒）。
total_fee	-	是	Number(64)	订单总金额，单位：分。
refund_fee	-	是	Number(64)	本次退款总金额，单位：分。
refund_fee_type	-	是	String(3)	本次退款总金额货币类型（目前只支持人民币，请填 CNY）。
refund_reason	-	否	String(255)	退款原因。
coupon_infos	-	否	CouponInfo[]	退款代金券信息，详见 CouponInfo。
is_deposit_mode	-	否	Bool	是否是押金消费单。
out_freeze_no	-	否	String(32)	此项为押金消费使用。 押金冻结单号。
freeze_fee	-	否	Number(32)	此项为押金消费使用。 押金冻结金额。
wxpay_refund_order_content_ext*	state	是	Number(32)	退款状态，详见 WxpayRefundOrderState。
	cash_refund_fee	否	Number(32)	现金退款金额，单位：分。
	coupon_refund_fee	否	Number(32)	退款代金券金额，支持多张代金券，单位：分。
	coupon_refund_count	否	Number(32)	退款代金券数量。
	refund_account	否	String(30)	退款资金来源，仅针对老资金流子商户使用。默认使用未结算资金退款： - REFUND_SOURCE_UNSETTLED_FUNDS：未结算资金退款。 - REFUND_SOURCE_RECHARGE_FUNDS：可用余额退款。
	refund_channel	否	String(16)	退款渠道。 - ORIGINAL：原路退款。 - BALANCE：退回到余额。
	refund_recv_account	否	String(64)	取当前退款单的退款入账方。 - 退回银行卡： {银行名称}{卡类型}{卡尾号} 。 - 退回支付用户零钱：支付用户零钱。

alipay_refund_order_content_ext*	refund_status	是	Number(32)	支付宝退款状态，详见枚举类型 AlipayRefundOrderState。
	fund_change	是	String(1)	是否发生了资金变化，示例 Y。
	gmt_refund_pay	是	String(32)	退款时间。
	refund_detail_item_list	否	AlipayFundBill	退款渠道，详见 AlipayFundBill。
record_refund_order_content_ext*	state	是	Number(32)	退款状态，详见枚举类型 RecordRefundOrderState。

带*项只会会有一个。

押金单信息

FreezeOrderContent 结构（仅作为返回参数）

一级参数名	二级参数名	必填	类型	说明
out_freeze_no	-	是	String	冻结押金授权单号。
freeze_id	-	否	String(255)	此项在押金消费时使用。第三方支付授权单号。
trade_type	-	是	Number(64)	支付方式。
author_code	-	否	String	刷卡支付授权码。
time_expire	-	否	Number(64)	订单失效时间。
time_end	-	否	Number(64)	支付完成时间。
create_time	-	是	Number(64)	冻结单创建时间。
last_update_time	-	是	Number(64)	冻结单创建时间。
is_transforming	-	是	Bool	系统是否正在处理。
freeze_fee	-	是	Number(32)	冻结金额，单位分。
fee_type	-	是	String	冻结货币类型。
cash_fee	-	是	Number(32)	现金支付金额（第三方支付平台返回）。
cash_fee_type	-	是	String	现金支付货币类型（第三方支付平台返回）。

settlement_total_fee	-	否	Number(32)	应结支付金额，单位分。
body	-	是	String(128)	商品或订单简要描述。
detail	-	否	String	商品详细列表，由 json 转化而来的 string，json 结构见本节 Detail。
remark	-	否	String(127)	支付备注信息。
total_consume_fee	-	否	Number(32)	冻结单消费金额，单位分。
total_manual_unfreeze_fee	-	否	Number(32)	冻结单解冻金额，单位分。
total_auto_unfreeze_fee	-	否	Number(32)	冻结单第三方自动解冻金额，单位分。
total_refund_fee	-	否	Number(32)	冻结消费后，退款金额，单位分。
wxpay_freeze_order_content_ext*	current_trade_state	是	Number(32)	当前微信支付冻结状态。
	attach	否	String(127)	附加数据，记录子商户自定义数据。
	bank_type	否	String(16)	付款银行类型，遵守字符型银行编码规范。
	trade_state_desc	否	String(255)	对当前查询冻结单状态的描述和下一步操作的指引。
	limit_pay	否	String(32)	限定支付方式，目前只能是：no_credit，指定不能使用信用卡支付。
	goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
alipay_freeze_order_content_ext*	current_trade_state	是	Number(32)	当前支付宝冻结状态。
	product_code	是	String(32)	销售产品码。
	enable_pay_channels	否	String	商户指定支付渠道。
	settle_fee_type	否	String(3)	结算 fee_type。

带*项只会会有一个。

UnfreezeOrderContent 结构（仅作为返回参数）

一级参数名	二级参数名	必填	类型	说明
out_freeze_no	-	是	String(32)	冻结单号。

out_unfreeze_no	-	是	String(32)	解冻单号。
unfreeze_id	-	否	String(32)	第三方解冻单信息。
trade_type	-	否	TradeType	交易类型，详见枚举 TradeType。
time_end	-	否	Number(64)	支付完成时间，时间戳（秒）。
nonce_str	-	否	Number(32)	随机字符串。
create_time	-	是	Number(64)	订单创建时间，时间戳（秒）。
last_update_time	-	是	Number(64)	订单最近更新时间，时间戳。
is_transforming	-	是	Bool	系统是否正在处理中。
unfreeze_fee	-	是	Number(32)	订单总金额（分）。
fee_type	-	否	String(3)	货币类型。
remark	-	否	String(127)	备注信息。
freeze_id	-	否	String(32)	第三方冻结单信息。
wxpay_unfreeze_order_content_ext*	current_trade_state	否	Number(32)	微信解冻单状态，详见枚举 WxpayUnfreezeOrderState。
alipay_unfreeze_order_content_ext*	current_trade_state	是	Number(32)	支付宝解冻单状态，详见枚举 AlipayUnfreezeOrderState。
	credit_amount	否	Number(64)	信用解冻金额。
	fund_amount	否	Number(64)	自有资金解冻金额。
	pre_auth_type	否	String(32)	预授权类型。

带*项只会有一个。

支付商户信息结构

PayMchKey 结构

一级参数名	二级参数名	必填	类型	说明
pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
sub_pay_platform	-	是	Number(32)	第三方支付子类型，详见枚举 SubPayPlatform。
out_mch_id	-	是	String(32)	服务商对外账号。
out_sub_mch_id	-	是	String(32)	子商户对外账号。

out_shop_id	-	是	String(32)	门店对外账号。
wxpay_pay_mch_key_ext*	app_id	否	String(32)	微信分配给服务商的公众账号 ID。
	sub_app_id	否	String(32)	微信分配给子商户的公众账号 ID。
alipay_pay_mch_key_ext*	user_id	否	String(32)	用户支付宝 user_id。

带*项只会会有一个。

OrderClient 结构

参数名	必填	类型	说明
shop_id	否	String(64)	子商户自定义，门店编号。
device_id	否	String(64)	子商户自定义，终端设备号。
staff_id	否	String(64)	子商户自定义，店员 ID。
terminal_type	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
sub_terminal_type	是	Number(32)	用户自定义子终端类型。
machine_no	否	String(32)	刷卡支付和刷卡押金冻结特有，每个收银终端的唯一码（刷卡支付和刷卡押金冻结必填）。
sdk_version	是	String(10)	云支付 SDK 版本号。
spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
sn_code	否	String(127)	设备 sn 号。

门店信息结构

ShopInfo 结构

参数名	必填	类型	说明
shop_id	是	String(64)	门店编号。
shop_name	是	String(32)	门店名称。
province	否	String(32)	门店所在省。
city	否	String(32)	门店所在市。
district	否	String(32)	门店所在区。
address	否	String(128)	门店详细地址。
coordinate_type	否	Number(32)	门店坐标类型。
longitude	否	String(16)	门店经度。
latitude	否	String(16)	门店维度。
height	否	String(16)	门店海拔。

phone	否	String(64)	门店联系电话。
out_shop_id	否	String(20)	外部可见的商户门店号，云支付内全局唯一，可用于生成门店固定二维码，仅返回时有该字段。
out_shop_id_url	否	String(128)	门店二维码，仅返回时有该字段。
device_infos	否	DeviceInfo[]	门店设备信息列表，详见 DeviceInfo。
staff_infos	否	StaffInfo[]	门店店员信息列表，详见 StaffInfo。
fee_type	否	String(20)	门店支持的币种，如果不填，默认为 CNY。

DeviceInfo 结构

参数名	必填	类型	说明
device_id	是	String(64)	门店内终端编号。
remark	否	String(64)	子商户自定义备注信息。
device_name	是	String	设备名称。
device_shift_type	否	Number(32)	设备类型，1：移动收款机具，2：云支付收银台，3：智能 POS，4：其它。

StaffInfo 结构

参数名	必填	类型	说明
staff_id	是	String(64)	门店内店员编号。
staff_name	是	String(64)	店员名称。
remark	否	String(64)	子商户自定义备注信息。
shop_manager	是	Bool	是否是店长。
receive_one_code_pay_notify	否	Bool	是否接收一码支付的成功消息通知。

CouponInfo 结构

参数名	必填	类型	说明
coupon_id	否	String(20)	代金券或立减优惠 ID 使用微信支付代金券时有返回。
coupon_fee	否	Number(64)	单个代金券或立减优惠支付金额。
coupon_type	否	String(8)	代金券类型，CASH：充值代金券，NO_CASH：非充值代金券，使用代金券时有返回。
contribute_type	否	Number(32)	资金来源，1：商户，2：平台，3：其他。

交易接口

刷卡支付

最近更新时间：2023-08-14 17:00:32

接口描述

项	值
url	https://pay.qcloud.com/cpay/micro_pay

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	author_code	-	是	String(256)	刷卡支付授权码。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	client_flow_id	-	否	String	客户端流水号。
	body	-	是	String(128)	商品或订单简要描述。 商品描述交易字段格式根据不同的应用场景按照以下格式： ● PC 网站：传入浏览器打开的网站主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 公众号：传入公众号名称-实际商品名称，例如：腾讯形象店-image-QQ 公仔。 ● H5：应用在浏览器网页上的场景，传入浏览器打开的移动网页的主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 线下门店：门店品牌名-城市分店名-实际商品名称，例如：image 形象店-深圳腾大-QQ 公仔。 ● App：需传入应用市场上的 App 名字-实际商品名称，天天爱消除-游戏充值。

	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。
	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext*	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
	sub_pay_platform	-	否	Number(32)	第三方支付子类型，详见枚举 SubPayPlatform。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
order_client	device_id	-	是	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	machine_no	-	是	String(32)	每个收银终端的唯一码。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	-	否	String(127)	设备 sn 号。
nonce_str	-	-	是	String(32)	随机字符串。

*只有微信支付时才使用。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见枚举值 Status。此字段仅表示接口调用成功状态，订单状态请以 current_trade_state 字段为准。
description	否	String(255)	错误描述。

log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
micro_pay	否	MicroPayResponse	authen_info 存在时必填。详见 MicroPayResponse。

MicroPayResponse 结构

参数名	必填	类型	说明
order_content	是	OrderContent	订单信息，详见公共数据结构 OrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
nonce_str	是	String(32)	随机字符串。

示例

输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "0AFB7AA87F09A295EDC5229470B82CC21345F0BBF1E69187948B9A92B6DCA183"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "sub_pay_platform": 100,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2I4B"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "order_client": {
      "staff_id": "1206",
      "machine_no": "1111",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "824",
      "spbill_create_ip": "90.0.00.0"
    },
    "pay_content": {
      "author_code": "134520273825387649",
      "body": "body",
      "fee_type": "CNY",
      "out_trade_no": "sz010002cz11566803216",
      "total_fee": 1
    }
  }"
}
```

输出参数示例：

```
{
  "response_content": "{
```

```

"status": 0,
"description": "\u64CD\u4F5C\u6210\u529F\u3002",
"log_id": 1310856473,
"internal_status": 0,
"micro_pay": {
  "pay_mch_key": {
    "pay_platform": 1,
    "out_mch_id": "sz01lXKA6DKGjNzr2l4B",
    "out_sub_mch_id": "sz01KzuCUOmW8yjtPite",
    "out_shop_id": "sz011biKxOguirmBqiFR",
    "sub_pay_platform": 100,
    "wxpay_pay_mch_key_ext": {
      "app_id": "wx998fa95558590e1c",
      "sub_app_id": "",
      "open_id": "ovJrDwsjamQql-k7Zr1Q-EaKFQ0k"
    }
  }
},
"order_content": {
  "out_trade_no": "sz010002cz11566803216",
  "transaction_id": "4200000400201908267240992395",
  "trade_type": 1,
  "author_code": "134520273825387649",
  "time_expire": 1566803336,
  "time_end": 1566803217,
  "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
  "create_time": 1566803216,
  "last_update_time": 1566803218,
  "is_transforming": false,
  "total_fee": 1,
  "fee_type": "CNY",
  "cash_fee": 1,
  "cash_fee_type": "CNY",
  "settlement_total_fee": 1,
  "body": "body",
  "detail": "",
  "wxpay_order_content_ext": {
    "current_trade_state": 2,
    "bank_type": "CFT",
    "goods_tag": "X293123",
    "coupon_fee": 0,
    "coupon_count": 0
  }
},
"nonce_str": "qB7ww6Yli0YqilpR7YgRPUnpW5A0nguj"
},
"authen_info": {
  "a": {
    "authen_type": 1,
    "authen_code": "252D91400A13C9D738A91C788D66987101E48BFF37AEB911AD7051A897C312EC"
  }
}
}

```

扫码支付

最近更新时间：2023-08-14 17:00:33

接口描述

项	值
url	https://pay.qcloud.com/cpay/scan_code_pay

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	time_expire	-	否	Number(64)	订单过期时间。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	client_flow_id	-	否	String	客户端流水号。
	body	-	是	String(128)	商品或订单简要描述。 商品描述交易字段格式根据不同的应用场景按照以下格式： ● PC 网站：传入浏览器打开的网站主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 公众号：传入公众号名称-实际商品名称，例如：腾讯形象店-image-QQ 公仔。 ● H5：应用在浏览器网页上的场景，传入浏览器打开的移动网页的主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 线下门店：门店品牌名-城市分店名-实际商品名称，例如：image 形象店-深圳腾大-QQ 公仔。 ● App：需传入应用市场上的 App 名字-实际商品名称，天天爱消除-游戏充值。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。

	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext*	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		product_id	否	String(32)	商品 ID，子商户自定义。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
	sub_pay_platform	-	否	Number(32)	第三方支付子类型，详见枚举 SubPayPlatform。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
order_client	device_id	-	是	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
	machine_no	-	是	String(32)	每个收银终端唯一码。
nonce_str	-	-	是	String(32)	随机字符串。

*只有微信支付时才使用。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0: 成功；非0: 失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。

scan_code_pay	否	ScanPayResponse	authen_info 存在时必填。详见 ScanPayResponse。
---------------	---	-----------------	---------------------------------------

ScanPayResponse 结构

参数名	必填	类型	说明
code_url	是	String	二维码 url。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "request_content": "{
    "nonce_str": "9f330667e18a48a7872e39878e1cd66f",
    "pay_content": {
      "out_trade_no": "sz0100d9ueWXzz00320190826151533",
      "wxpay_pay_content_ext": {
        "product_id": "product_id"
      },
      "total_fee": 16000,
      "fee_type": "CNY",
      "body": "body"
    },
    "order_client": {
      "machine_no": "01-01-01-01-01-01",
      "terminal_type": 2,
      "device_id": "zz003",
      "staff_id": "zz003",
      "sdk_version": "1.2",
      "spbill_create_ip": "192.168.0.22",
      "sub_terminal_type": 900
    },
    "pay_mch_key": {
      "out_mch_id": "sz01N6YTRV8JPllb4AnB",
      "out_shop_id": "sz01Kmw5JHJs278376ndk",
      "pay_platform": 1,
      "out_sub_mch_id": "sz01jRnVDGfHKHI837nH"
    }
  }",
  "authen_info": {
    "a": {
      "authen_code": "608BB7BD002E5A57F5EF501965EBD540B8039CB51916D1F3727F7A74DCADAE0B",
      "authen_type": 1
    }
  }
}
```

- 输出参数示例：

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 4049520517,
    "internal_status": 0,
  }
```



```
"scan_code_pay": {  
  "code_url": "weixin:\\\\wxpay\\bizpayurl?pr=3oi3lB6",  
  "nonce_str": "beM31vIaC1rgyDIX43fjxjhs77eKMrlf"  
}  
},  
"authen_info": {  
  "a": {  
    "authen_type": 1,  
    "authen_code": "57946ECEED1554D7B95D6650B42701EA0DAB6292A3CC79D4E4C7C5F86EA1DDC1"  
  }  
}  
}
```

公众号支付

最近更新时间：2023-08-17 10:35:11

接口描述

项	值
url	https://pay.qcloud.com/cpay/official_account_pay

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	time_expire	-	否	Number(64)	订单过期时间，秒级时间戳。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	body	-	是	String(128)	商品或订单简要描述。 商品描述交易字段格式根据不同的应用场景按照以下格式： ● PC 网站：传入浏览器打开的网站主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 公众号：传入公众号名称-实际商品名称，例如：腾讯形象店-image-QQ 公仔。 ● H5：应用在浏览器网页上的场景，传入浏览器打开的移动网页的主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 线下门店：门店品牌名-城市分店名-实际商品名称，例如：image 形象店-深圳腾大-QQ 公仔。 ● App：需传入应用市场上的 App 名字-实际商品名称，天天爱消除-游戏充值。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。

	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext[1]	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		product_id	否	String(32)	商品 ID，子商户自定义。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
	sub_pay_platform	-	否	Number(32)	第三方支付子类型，详见枚举 SubPayPlatform。
	out_mch_id	-	是	String(32)	服务商对外帐号。
	out_sub_mch_id	-	是	String(32)	子商户对外帐号。
	out_shop_id	-	是	String(32)	门店对外帐号。
	wxpay_pay_mch_key_ext[2]	app_id	是	String(32)	微信分配给服务商的公众账号 ID。
		sub_app_id	否	String(32)	微信分配给子商户的公众账号 ID。
		open_id	是	String(128)	用户在服务商 app_id 下的唯一标识
		sub_open_id	否	String(128)	用户在子商户 app_id 下的唯一标识。
	alipay_pay_mch_key_ext[2]	user_id	是	String(32)	用户支付宝 user_id。
order_client	device_id	-	否	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	否	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	-	否	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
nonce_str	-	-	是	String(32)	随机字符串。

- [1]: 只有微信支付时才使用。
- [2]: wxpay_pay_mch_key_ext 和 alipay_pay_mch_key_ext 只需要填写一种。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，该 string 可以转为 JSON 结构，JSON 格式见本节 ResponseContent。

authen_info	否	AuthenInfo	认证信息，详见接口调用说明。
-------------	---	------------	----------------

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
official_account_pay	否	OfficialAccountResponse	authen_info 存在时必填。详见 OfficialAccountPayResponse。

OfficialAccountPayResponse 结构

参数名	必填	类型	说明
order_content	是	OrderContent	订单信息，详见公共数据结构 OrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
jsapi_args	是	JsapiArguments	JSAPI 支付需要的参数，详见公共数据结构 JsapiArguments。
nonce_str	是	String(32)	随机字符串。

示例

• 输入参数示例：

```
{
  "request_content": "{
    "pay_mch_key": {
      "out_mch_id": "sz01CRXVcdYDyJik8873",
      "out_sub_mch_id": "sz01KuceBtrckIKw339L",
      "out_shop_id": "sz01NmziKgZQmAl98Kij",
      "pay_platform": 1,
      "wxpay_pay_mch_key_ext": {
        "sub_open_id": "oxEiL1Yq1ellkl987C-o3v9y6MYQ"
      }
    },
    "pay_content": {
      "out_trade_no": "sz017439399162737843772384118",
      "total_fee": 10000,
      "fee_type": "CNY",
      "body": "\u6e56\u5357\u4ea4\u901a\u5de5\u7a0b\u5b66\u9662\u540e\u52e4\u5904"
    },
    "order_client": {
      "spbill_create_ip": "192.168.1.1"
    },
    "nonce_str": "57vJCSbipmT0z00TJhw9xIHn3QGdZBah",
    "trade_type": 3
  }",
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "503EB8AE9F3CB922F1F772FAE0C60247BB5CE50BC84050723F154A5FEE64132C"
    }
  }
}
```

```
}  
}
```

• 输出参数示例:

```
{  
  "response_content": "{  
    "status": 0,  
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",  
    "log_id": 3056348287,  
    "internal_status": 0,  
    "official_account_pay": {  
      "pay_mch_key": {  
        "pay_platform": 1,  
        "out_mch_id": "sz01CRXVcdYDyT5TC38H",  
        "out_sub_mch_id": "sz01KuceBtrcksX1OW6V",  
        "out_shop_id": "sz01NmziKgZQmaT07BQI",  
        "sub_pay_platform": 100,  
        "wxpay_pay_mch_key_ext": {  
          "app_id": "wx929d56168e0xxxxx",  
          "sub_app_id": "wx0217eb58c7xxxxxx",  
          "sub_open_id": "oxEiL1Yq1eIlkl987C-o3v9y6MYQ "  
        }  
      },  
      "order_content": {  
        "out_trade_no": "sz010187m90829162737843772384118",  
        "trade_type": 3,  
        "time_expire": 1567067378,  
        "nonce_str": "57vJCSbipmT0z00TJhw9xIHn3QGdZBah",  
        "create_time": 1567067258,  
        "last_update_time": 1567067258,  
        "is_transforming": false,  
        "total_fee": 10000,  
        "fee_type": "CNY",  
        "cash_fee_type": "CNY",  
        "body": "\u6E56\u5357\u4EA4\u901A\u5DE5\u7A0B\u5B66\u9662\u540E\u52E4\u5904",  
        "wxpay_order_content_ext": {  
          "current_trade_state": 6,  
          "goods_tag": "",  
          "prepare_id": "wx2916273827558523ae1a04971089444300"  
        }  
      },  
      "nonce_str": "K3hIsvsZEffiLrm8n2HP2wx2IPZzNkUU",  
      "jsapi_args": {  
        "wxpay": {  
          "get_brand_wcpay": {  
            "app_id": "wx369ab4160391b078",  
            "timestamp": "1567067258",  
            "nonce_str": "5cee0f921bc24254140045e4fe22483b",  
            "package": "prepay_id=wx2916273827558523ae1a04971089444300",  
            "sign_type": "MD5",  
            "pay_sign": "35E6E610203BCC85603B0A2C7996BB03"  
          }  
        }  
      }  
    },  
    "authen_info": {  
      "a": {
```

```
"authen_type": 1,  
"authen_code": "9AFBB5A348A35F5696D46E8BDDD8716A5750FD14515C00BAC1E3A60E4419D46E"  
}  
}  
}
```

H5 支付

最近更新时间：2023-08-17 10:35:11

接口描述

手机 H5 支付接口，目前仅支持微信支付。

项	值
url	https://pay.qcloud.com/cpay/mob_web_pay

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	time_expire	-	否	Number(64)	订单过期时间。
	body	-	是	String(128)	商品或订单简要描述，需传入应用市场上的“APP 名称-实际商品名称”，如“天天爱消除-游戏充值”。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。
	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
		scene_info	是	String(256)	支付的场景信息 json 串，参数格式参考微信官方文档 参数说明部分 。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。

	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
order_client	spbill_create_ip	-	是	String(16)	客户端 IP。
nonce_str	-	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，该 string 可以转为 json 结构，json 格式见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 枚举值定义-Status 枚举变量 。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
mobile_web_pay	否	MobileWebPayResponse	authen_info 存在时必填。详见 MobileWebPayResponse。

MobileWebPayResponse 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	-	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
nonce_str	-	是	String(32)	随机字符串。
mobile_web_pay	wxpay	否	WxpayMWebArguments	微信 H5 支付 SDK 接口参数，结构见下文。

WxpayMWebArguments 结构

参数名	必填	类型	说明
mweb_url	是	String	浏览器用来拉起微信 H5 支付的 URL。可在 mweb_url 后拼接参数 redirect_url，指定回调页面。当用户支付完成后，手机浏览器会调整到 redirect_url。
prepay_id	是	String	微信返回的支付交易会话 ID，目前没有实际作用。

示例

- 输入参数示例：

```
{
  "authen_info": {
```



```

    "a": {
      "authen_type": 1,
      "authen_code": "2D51EDAB82771A999B80F8EEE38CEC1A68FC2BDC9BCCA12C0B6EDA8BC22BF5DA"
    },
    "request_content": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01fdd7icm7JKjkkfc3",
        "out_sub_mch_id": "sz01LG0dvp4O09jjuPbA",
        "out_shop_id": "sz011Zszd73bocifl9eG"
      },
      "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
      "order_client": {
        "spbill_create_ip": "90.0.00.0"
      },
      "pay_content": {
        "body": "body",
        "out_trade_no": "010000521567668680",
        "wxpay_pay_content_ext": {
          "scene_info": {"h5_info": {"type": "Wap", "wap_url": "https://pay.qq.com\\", "wap_name": "\\u817e\\u8baf\\u5145\\u503c"}}
        },
        "total_fee": 100,
        "fee_type": "CNY"
      }
    }
  }
}

```

● 输出参数示例:

```

{
  "response_content": {
    "status": 0,
    "description": "\\u64CD\\u4F5C\\u6210\\u529F\\u3002",
    "log_id": 3637968353,
    "internal_status": 0,
    "mobile_web_pay": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01fdd7icm7JKjkkfc3",
        "out_sub_mch_id": "sz01LG0dvp4O09jjuPbA",
        "out_shop_id": "sz011Zszd73bocifl9eG",
        "sub_pay_platform": 100
      },
      "nonce_str": "zWbxyYBupz2DDIRiO9u9BsmjbAQsmnGv",
      "mweb_args": {
        "wxpay": {
          "mweb_url": "https://wx.tenpay.com/cgi-bin/mmpayweb-bin/checkmweb?prepay_id=wx2016121516420242444321ca0631331346&package=1405458241",
          "prepay_id": "wx201411101639507cbf6ffd8b0779950874"
        }
      }
    }
  },
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "852F74DA744049B1C629056FE8E8CC7BB59C49A25546ED66C8B75F4E6475AD73"
    }
  }
}

```

```
    }  
  }  
}
```

APP 支付

最近更新时间：2023-08-17 10:35:11

接口描述

手机 App 支付接口，目前仅支持微信支付。

项	值
url	https://pay.qcloud.com/cpay/app_pay

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	time_expire	-	否	Number(64)	订单过期时间。
	body	-	是	String(128)	商品或订单简要描述，需传入应用市场上的“APP 名称-实际商品名称”，如“天天爱消除-游戏充值”。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。
	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。

	out_shop_id	-	是	String(32)	门店对外账号。
	wxpay_pay_mch_key_ext	mob_app_id	是	String(32)	受理商户的 APPID，在微信开放平台绑定申请。
nonce_str	-	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
app_pay	否	AppPayResponse	authen_info 存在时必填。详见 AppPayResponse。

AppPayResponse 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	-	是	PayMchKey	支付商户信息，详见公共结构 PayMchKey。
nonce_str	-	是	String(32)	随机字符串。
sdk_args	wxpay	否	WxpaySdkArguments	微信 App 支付 SDK 接口参数，结构见下文。

WxpaySdkArguments 结构

参数名	必填	类型	说明
app_id	是	String	子商户在微信开放平台上申请的 APPID。
timestamp	是	String	时间戳。
nonce_str	是	String	随机字符串。
package_value	是	String	暂填写固定值 Sign=WXPAY。
partner_id	是	String	子商户的商户号。
sign	是	String	签名。
prepare_id	是	String	微信返回的支付交易会话 ID。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "E6E70FF7E330E046239B05E100942B15C469928B044DF6488F3BD88AB385C682"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "out_sub_mch_id": "sz01LG0dvp4O09jjuPbA",
      "wxpay_pay_mch_key_ext": {
        "mob_app_id": "wx87dd198b78jucir8"
      },
      "out_mch_id": "sz01fdd7icm7JKjkkfc3",
      "out_shop_id": "sz011Zszd73bocifl9eG",
      "pay_platform": 1,
      "out_mch_id": "sz01MaHltuPggtcvcTUM"
    },
    "nonce_str": "k8s03nyOPIQxju4lw69CCfeUWL66GDBH",
    "pay_content": {
      "body": "body",
      "fee_type": "CNY",
      "out_trade_no": "011888520000211567600739877620",
      "total_fee": 100,
      "time_expire": 1567601636
    }
  }"
}
```

● 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 3637968353,
    "internal_status": 0,
    "app_pay": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01fdd7icm7JKjkkfc3",
        "out_sub_mch_id": "sz01LG0dvp4O09jjuPbA",
        "out_shop_id": "sz011Zszd73bocifl9eG",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx0b0ebc671d51405b",
        }
      },
      "nonce_str": "zWbxyYBupz2DDIRiO9u9BsmjbAQsmnGv",
      "sdk_args": {
        "wxpay": {
          "app_id": "wx87dd198b78jucir8",
          "partner_id": "257300488",
          "package_value": "Sign=WXPay",
          "nonce_str": "k8s03nyOPIQxju4lw69CCfeUWL66GDBH",
          "timestamp": "1567600739",
          "sign": "B3E62ECF398573085397AFCAE1568A8A"
        }
      }
    }
  }
}
```

```
}  
}",  
"authen_info": {  
  "a": {  
    "authen_type": 1,  
    "authen_code": "852F74DA744049B1C629056FE8E8CC7BB59C49A25546ED66C8B75F4E6475AD73"  
  }  
}  
}
```

申请退款

最近更新时间：2023-08-14 17:00:33

接口描述

项	值
url	https://pay.qcloud.com/cpay/refund

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
refund_content	out_trade_no	-	是	String(32)	订单号。
	out_refund_no	-	是	String(32)	退款单号。
	total_fee	-	是	Number(32)	订单总金额，金额分。
	refund_fee	-	是	Number(32)	本次退款金额，金额分。
	refund_fee_type	-	是	String(3)	本次退款货币类型。
	refund_reason	-	否	String(64)	退款原因备注。
	client_refund_flow_id	-	否	Number(64)	客户端流水号。
	out_freeze_no	-	否	String(32)	押金冻结单号（押金模式退款单必填）。
	wx_pay_refund_content_ext*	refund_account	否	String(30)	退款资金来源，仅针对老资金流子商户使用。默认使用未结算资金退款。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
	is_sub_mch_admin	-	是	Bool	操作者是否为商户管理员（此参数是为了校验退款人身份，操作退款人必须拥有退款权限。true 代表操作者是商户管理员，该角色默认拥有退款权限。当为 false 时

					与 authorization_staff_id 配合使用校验退款权限)。
	authorization_staff_id	-	否	String(64)	如果 is_sub_mch_admin 为 false, 则此项为必填, 如果为 true, 则此项不填。1. 如果当前 staff_id 有退款权限则传自身的 staff_id。2. 如果是其他人授权则填授权者的 staff_id。3. 机具调用传机具内配置的 staff_id。
	authorization_name	-	否	String(32)	如果当前操作者有退款权限则传自身的 name, 如果是其他人授权则填授权者的 name。此字段最终会补充在返回结构的退款原因 (refund_reason) 字段中。
order_client	device_id	-	否	String(64)	子商户自定义, 终端设备号。
	terminal_type	-	是	Number(32)	终端类型, 1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	-	否	String(127)	设备 sn 号。
api_version	-	-	否	Number(32)	默认为1。如果填1, 则不认证操作者退款权限; 如果填2, 则需要结合 pay_mch_key 中的 is_sub_admin, authorization_staff_id 字段验证退款权限。
nonce_str	-	-	是	String(32)	随机字符串。

*只有微信支付时才使用。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容, 详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息, 详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码, 详见 Status。0: 成功; 非0: 失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用, 调用者可以不予理会。
refund	否	RefundResponse	authen_info 存在时必填。详见 RefundResponse。

RefundResponse 结构

参数名	必填	类型	说明
pay_mch_key	是	PayMchkey	支付商户信息，status 为0时必填。详见公共数据结构 PayMchKey。
refund_order_content	否	RefundOrderContent	退款单信息，详见公共数据结构 RefundOrderContent。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "0AFAFB3BF238BCD1BD2617C9AE720FA83D600911BA3E6FD95D62A5EED90569D6"
    }
  },
  "request_content": "{
    \"refund_content\": {
      \"out_refund_no\": \"sz010002cz11571214438\",
      \"out_trade_no\": \"sz010002cz11571214366\",
      \"refund_fee_type\": \"CNY\",
      \"total_fee\": 1,
      \"refund_reason\": \"returns\",
      \"refund_fee\": 1
    },
    \"pay_mch_key\": {
      \"pay_platform\": 1,
      \"out_mch_id\": \"sz01IXKA6DKGjNzr2l4B\",
      \"out_sub_mch_id\": \"sz01KzuCUOmW8yjtPite\",
      \"out_shop_id\": \"sz011biKxOguirmBqiFR\",
      \"is_sub_mch_admin\": false,
      \"authorization_staff_id\": \"123123\",
      \"authorization_name\": \"dianyuan\"
    },
    \"order_client\": {
      \"terminal_type\": 1,
      \"sdk_version\": \"1.0\",
      \"spbill_create_ip\": \"90.0.00.0\"
    },
    \"api_version\": 2,
    \"nonce_str\": \"E94C00688C3F429CA2B0B396BF823548\"
  }"
```

- 输出参数示例:

```
{
  "response_content": "{
    \"status\": 0,
    \"description\": \"\u64CD\u4F5C\u6210\u529F\u3002\",
    \"log_id\": 4237349548,
    \"internal_status\": 0,
    \"refund\": {
```

```
"pay_mch_key": {
  "pay_platform": 1,
  "out_mch_id": "sz01lXKA6DKGjNzr2l4B",
  "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
  "out_shop_id": "sz011biKxOguirmBqiFR",
  "sub_pay_platform": 100,
  "wxpay_pay_mch_key_ext": {
    "app_id": "wx998fa95558590s21",
  }
},
"refund_order_content": {
  "out_refund_no": "sz010002cz11571214438",
  "refund_id": "50000702232019101612463710837",
  "trade_type": 1,
  "out_trade_no": "sz010002cz11571214366",
  "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
  "create_time": 1571214438,
  "last_update_time": 1571214440,
  "refund_reason": "returns\n\u64CD\u4F5C\u8005\uFF1A1206\n\u6388\u6743\u8005\uFF1A\u5927\u5B9D",
  "total_fee": 1,
  "refund_fee": 1,
  "refund_fee_type": "CNY",
  "wxpay_refund_order_content_ext": {
    "state": 4,
    "cash_refund_fee": 1,
    "coupon_refund_fee": 0,
    "coupon_refund_count": 0
  },
  "is_deposit_mode": false,
},
"nonce_str": "ymT9uMh4CV7ZrgESjG2brAlb8a1l1knu"
}
},
"authen_info": {
  "a": {
    "authen_type": 1,
    "authen_code": "C4C73130FE6CBF1CF185909057067B622EBBF55E95339886A18024D2568CC699"
  }
}
}
```

查询订单

最近更新时间：2023-08-14 17:00:33

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_order

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	pay_platform	否	Number(32)	第三方支付类型。
	sub_pay_platform	否	Number(32)	第三方支付子类型。
	out_mch_id	是	String(32)	服务商对外账号。
	out_sub_mch_id	是	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
order_client	device_id	否	String(64)	子商户自定义，终端设备号。
	staff_id	否	String(64)	子商户自定义，店员 ID。
	terminal_type	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	否	String(127)	设备 sn 号。
out_trade_no	-	是	String(32)	订单号。
trade_type	-	否	Number(32)	支付方式，详见枚举 TradeType。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见枚举值 Status。此字段仅表示接口调用成功状态，订单状态请以 current_trade_state 字段为准。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_order	否	QueryOrderResponse	authen_info 存在时必填。详见 QueryOrderResponse。

QueryOrderResponse 结构

参数名	必填	类型	说明
order_content	是	OrderContent	订单信息，详见公共数据结构 OrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
order_client	否	OrderClient	详见公共数据结构 OrderClient。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "664A939ECD36280AE1E6D39029011216CEF4A1DDE1640098F6FED7E88CFD707D"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "sub_pay_platform": 100,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmW8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2I4B"
    },
    "trade_type": 1,
    "out_trade_no": "sz010002cz11566803216",
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "order_client": {
      "staff_id": "1192",
      "machine_no": "1111",
      "terminal_type": 1,

```

```
"sdk_version": "1.0",
"device_id": "12345",
"spbill_create_ip": "90.0.0.0"
}
}"
}
```

• 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 2817472575,
    "internal_status": 0,
    "query_order": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
        "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
        "out_shop_id": "sz011biKxOguirmBqiFR",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx998fa95558590e1c",
          "sub_app_id": "",
          "open_id": "ovJrDwsjamQql-k7Zr1Q-EaKFQ0k",
          "sub_open_id": ""
        }
      }
    },
    "order_content": {
      "out_trade_no": "sz010002cz11566803216",
      "transaction_id": "4200000400201908267240992395",
      "trade_type": 1,
      "author_code": "134520273825387649",
      "time_expire": 1566803336,
      "time_end": 1566803217,
      "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
      "create_time": 1566803216,
      "last_update_time": 1566803218,
      "is_transforming": false,
      "source_type": 0,
      "caller_notify_url": "",
      "total_fee": 1,
      "fee_type": "CNY",
      "cash_fee": 1,
      "cash_fee_type": "CNY",
      "settlement_total_fee": 1,
      "body": "body",
      "detail": "",
      "context": "",
      "exchange_rate": 100000000,
      "client_flow_id": "",
      "remark": "",
      "trade_state": 4,
      "wxpay_order_content_ext": {
        "current_trade_state": 2,
        "attach": "",
        "bank_type": "CFT",
        "goods_tag": "X293123",

```

```
        "coupon_fee": 0,
        "coupon_count": 0,
        "product_id": "",
        "limit_pay": "",
        "prepare_id": "",
        "trade_state_desc": ""
    }
},
"nonce_str": "TqSHJKFpAlCOr3nfKWf4QTKwmiboi0Sn",
"order_client": {
    "device_id": "824",
    "staff_id": "1206",
    "machine_no": "1111",
    "terminal_type": 1,
    "sdk_version": "1.0",
    "spbill_create_ip": "90.0.00.0",
    "sub_terminal_type": 0
}
},
"authen_info": {
    "a": {
        "authen_type": 1,
        "authen_code": "428C7A16E605C21CF1A939882D7A195D26E041A3CAEFF004F11B0B1CC7EE8727"
    }
}
}
```

关闭订单

最近更新时间：2023-08-17 10:35:12

接口描述

项	值
url	https://pay.qcloud.com/cpay/close_order

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	pay_platform	是	Number(32)	第三方支付类型。
	sub_pay_platform	否	Number(32)	第三方支付子类型。
	out_mch_id	是	String(32)	服务商对外账号。
	out_sub_mch_id	是	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
order_client	device_id	是	String(64)	子商户自定义，终端设备号。如果是公众号支付订单，则此项可不填写。
	staff_id	否	String(64)	子商户自定义，店员 ID。
	terminal_type	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	否	String(127)	设备 sn 号。
out_trade_no	-	是	String(32)	订单号。
trade_type	-	是	Number(32)	支付方式，详见枚举 TradeType。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，该 string 可以转为 json 结构，json 格式见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
close_order	否	CloseOrderResponse	authen_info 存在时必填。详见 CloseOrderResponse。

CloseOrderResponse 结构

参数名	必填	类型	说明
nonce_str	是	String(32)	随机字符串

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "664A939ECD36280AE1E6D39029011216CEF4A1DDE1640098F6FED7E88CFD707D"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "sub_pay_platform": 100,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B"
    },
    "trade_type": 1,
    "out_trade_no": "sz010002cz11566803216",
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "order_client": {
      "staff_id": "1192",
      "machine_no": "1111",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "12345",
      "spbill_create_ip": "90.0.00.0"
    }
  }"
}
```


- 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 2817472575,
    "internal_status": 0,
    "query_order": {
      "nonce_str": "TqSHJKFpAlCO3nfKWf4QTKwmiboi0Sn"
    }
  }",
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "428C7A16E605C21CF1A939882D7A195D26E041A3CAEFF004F11B0B1CC7EE8727"
    }
  }
}
```

查询退款单

最近更新时间：2023-08-14 17:00:33

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_refund_order

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	pay_platform	是	Number(32)	第三方支付类型。
	sub_pay_platform	否	Number(32)	第三方支付子类型。
	out_mch_id	是	String(32)	服务商对外账号。
	out_sub_mch_id	是	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
order_client	device_id	否	String(64)	子商户自定义，终端设备号。
	terminal_type	是	Number(32)	终端类型，1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
out_trade_no	-	是	String(32)	订单号。
out_refund_no	-	是	String(32)	退款单号。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。

authen_info	否	AuthenInfo	认证信息，详见接口调用说明。
-------------	---	------------	----------------

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_refund_order	否	QueryRefundOrderResponse	authen_info 存在时必填。详见 QueryRefundOrderResponse。

QueryRefundOrderResponse 结构

参数名	必填	类型	说明
pay_mch_key	是	PayMchKey	支付商户信息，status 为0时必填。详见公共数据结构 PayMchKey。
refund_order_content	否	RefundOrderContent[]	退款单信息列表，详见公共数据结构 RefundOrderContent。
order_client	否	OrderClient	详见公共数据结构 OrderClient。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "1068AA0F2CBEBEAEC3DA1115C32EBE54B303A65A64264BCA1D89564E1D9F004D"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "sub_pay_platform": 100,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2I4B"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "out_trade_no": "sz010002cz11566803216",
    "out_refund_no": "sz010002cz11566825386",
    "order_client": {
      "staff_id": "1192",
      "machine_no": "1111",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "12345",
      "spbill_create_ip": "90.0.00.0"
    }
  },
  "refund_order_content": [
    {
      "refund_order_no": "sz010002cz11566803216",
      "refund_order_amount": 100,
      "refund_order_status": 0,
      "refund_order_desc": "退款单描述"
    }
  ]
}
```

- **输出参数示例:**

版权所有：腾讯云计算（北京）有限责任公司

```
        "sdk_version": "1.0",
        "spbill_create_ip": "90.0.00.0",
        "sub_terminal_type": 0
    }
}
},
"authen_info": {
    "a": {
        "authen_type": 1,
        "authen_code": "18C23A91D494E809EB34C392B2EC5B9CAD4D5B9BD716FB0E14F805278B0BD36F"
    }
}
}
```

支付成功回调

最近更新时间：2020-06-29 15:10:13

接口描述

- 此接口为云支付系统主动调用服务商后台系统的请求。
- 支付成功回调的 URL 地址是服务商在控制台配置的“交易完成回调 URL”，**并非支付接口参数中的 `caller_notify_url` **。
- 如果云支付后台接收到服务商系统回包失败，至多会重试三次，每次间隔30秒 - 60秒。

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息， authen_code 生成使用的是服务商的 authen_key 进行签名计算，详见接口调用说明。

RequestContent 结构

参数名	必填	类型	说明
nonce_str	是	String(32)	随机字符串。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
order_content	是	OrderContent	订单信息，详见公共数据结构 OrderContent。
order_client	是	OrderClient	订单客户端信息，详见公共数据结构 OrderClient。

输出参数

无需返回内容。云支付系统根据返回的 HTTP 状态码做如下操作：HTTP 状态码为200，云支付系统认为消息正常接受，否则会发起重试，最多重试三次。

示例

输入参数示例：

```
{
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "wxpay_pay_mch_key_ext": {
        "app_id": "wx998fa95558590e1c",
        "open_id": "ovJrDwsjamQql-k7Zr1Q-EaKFQ0k",
      }
    },
    "order_content": {
      "out_trade_no": "sz010002cz11566803216",
      "transaction_id": "4200000400201908267240992395",
      "trade_type": 1,
      "author_code": "134520273825387649",
      "time_expire": 1566803336,
    }
  }
}
```

```
"time_end": 1566803217,
"nonce_str": "E94C00688C3F429CA2B0B396BF823548",
"create_time": 1566803216,
"last_update_time": 1566803218,
"is_transforming": false,
"source_type": 0,
"caller_notify_url": "",
"total_fee": 1,
"fee_type": "CNY",
"cash_fee": 1,
"cash_fee_type": "CNY",
"settlement_total_fee": 1,
"body": "body",
"detail": "",
"context": "",
"exchange_rate": 100000000,
"trade_state": 4,
"wxpay_order_content_ext": {
  "current_trade_state": 2,
  "attach": "",
  "bank_type": "CFT",
  "goods_tag": "X293123",
  "coupon_fee": 0,
  "coupon_count": 0,
  "product_id": "",
  "limit_pay": "",
  "prepare_id": "",
  "trade_state_desc": ""
},
},
"nonce_str": "TqSHJKFpAlCOR3nfKWf4QTKwmiboi0Sn",
"order_client": {
  "device_id": "824",
  "staff_id": "1206",
  "machine_no": "1111",
  "terminal_type": 1,
  "sdk_version": "1.0",
  "spbill_create_ip": "90.0.00.0",
  "sub_terminal_type": 0
}
},
"authen_info": {
  "a": {
    "authen_type": 1,
    "authen_code": "428C7A16E605C21CF1A939882D7A195D26E041A3CAEFF004F11B0B1CC7EE8727"
  }
}
}
```

刷卡押金冻结

最近更新时间：2023-08-14 17:00:33

接口描述

项	值
url	https://pay.qcloud.com/cpay/micro_freeze

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
freeze_content	out_freeze_no	-	是	String(32)	冻结押金授权单号。
	author_code	-	是	String(256)	刷卡支付授权码。
	freeze_fee	-	是	Number(32)	冻结金额，单位：分。
	fee_type	-	是	String	冻结货币类型。
	body	-	是	String(128)	商品或订单简要描述。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共结构 Detail。
	remark	-	否	String	支付备注信息。
	wxpay_freeze_content_ext[1]	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
	alipay_freeze_content_ext[1]	enable_pay_channels	否	String	商户指定支付渠道。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型。

	sub_pay_platform	-	否	Number(32)	第三方支付子类型。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
	wxpay_pay_mch_key_ext[2]	app_id	否	String(32)	微信分配给服务商的公众账号 ID。
		sub_app_id	否	String(32)	微信分配给子商户的公众账号 ID。
	alipay_pay_mch_key_ext[2]	user_id	否	String(32)	用户支付宝 user_id。
order_client	device_id	-	是	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	是	Number(32)	终端类型，1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	machine_no	-	是	String(32)	每个收银终端的唯一码。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	-	否	String(127)	设备 sn 号。
nonce_str	-	-	是	String(32)	随机字符串。

注意

- 1. wxpay_freeze_content_ext 与 alipay_freeze_content_ext 只能填写一个。
- 2. wxpay_pay_mch_key_ext 与 alipay_pay_mch_key_ext 只能填写一个。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见枚举值 Status。此字段仅表示接口调用成功状态，订单状态请以 current_trade_state 字段为准。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。

internal_status	是	Number(32)	调试使用，调用者可以不予理会。
micro_freeze	否	MicroFreezeResponse	authen_info 存在时必填。详见 MicroFreezeResponse。

MicroFreezeResponse 结构

参数名	必填	类型	说明
freeze_order_content	是	FreezeOrderContent	冻结单信息，详见公共数据结构 FreezeOrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "35EAA19538EFD9A9A2A22D48AF262F09B774986251231F8F45DA8C7DBF8275D3"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "sub_pay_platform": 100,
      "out_shop_id": "1234ruQCleTa9w30AaAH",
      "out_sub_mch_id": "12343ycHpBDv8GXjfmSvT7",
      "out_mch_id": "1234mcWYS3iM5TjKLorAZ"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "freeze_content": {
      "author_code": "32323232",
      "body": "body",
      "out_freeze_no": "sz010002cz11564384955",
      "freeze_fee": 10,
      "fee_type": "CNY"
    },
    "order_client": {
      "staff_id": "1206",
      "machine_no": "00-05-9a-3c-7a-00",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "824",
      "spbill_create_ip": "90.0.00.0"
    }
  }"
}
```

- 输出参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
```

```
    "authen_code": "3B11C7489804E267C917DBA8508D403A3CF53C895272AC43587A631325629A18"
  }
},
"response_content": "{
  "status": 0,
  "description": "\u64CD\u4F5C\u6210\u529F\u3002",
  "log_id": 1971094428,
  "internal_status": 0,
  "micro_freeze": {
    "pay_mch_key": {
      "pay_platform": 1,
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "sub_pay_platform": 100,
      "wxpay_pay_mch_key_ext": {
        "app_id": "wx998fa95558590e1c",
        "open_id": "ovJrDwsjamQql-k7Zr1Q-EaKFQ0k"
      }
    }
  },
  "freeze_order_content": {
    "out_freeze_no": "sz010002cz11564386781",
    "trade_type": 1,
    "author_code": "134550264496882447",
    "time_expire": 1564386901,
    "time_end": 1564386782,
    "create_time": 1564386781,
    "last_update_time": 1564386783,
    "is_transforming": false,
    "freeze_fee": 10,
    "fee_type": "CNY",
    "cash_fee": 10,
    "cash_fee_type": "CNY",
    "settlement_total_fee": 10,
    "body": "body",
    "wxpay_freeze_order_content_ext": {
      "current_trade_state": 2,
      "bank_type": "CFT",
      "goods_tag": "X293123"
    }
  },
  "nonce_str": "hFiSO4JaMpDB38qWELxmG45Maue5Wdzp"
}"
}
```

扫码押金冻结

最近更新时间：2023-08-14 17:00:33

接口描述

扫码押金冻结当前仅支持支付宝。

项	值
url	https://pay.qcloud.com/cpay/scancode_freeze

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
freeze_content	out_freeze_no	-	是	String(32)	冻结押金授权单号。
	freeze_fee	-	是	Number(32)	冻结金额，单位：分。
	fee_type	-	是	String	冻结货币类型。
	time_expire	-	否	Number(64)	订单过期时间。
	body	-	是	String(128)	商品或订单简要描述。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共结构 Detail。
	remark	-	否	String	支付备注信息。
	wxpay_freeze_content_ext	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
	alipay_freeze_content_ext	enable_pay_channels	否	String	商户指定支付渠道。

pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型。
	sub_pay_platform	-	否	Number(32)	第三方支付子类型。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
	wxpay_pay_mch_key_ext[1]	app_id	否	String(32)	微信分配给服务商的公众账号 ID。
		sub_app_id	否	String(32)	微信分配给子商户的公众账号 ID。
	alipay_pay_mch_key_ext[1]	user_id	否	String(32)	用户支付宝 user_id。
order_client	device_id	-	是	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
nonce_str	-	-	是	String(32)	随机字符串。

 注意

- wxpay_freeze_content_ext 与 alipay_freeze_content_ext 只能填写一个。
- wxpay_pay_mch_key_ext 与 alipay_pay_mch_key_ext 只能填写一个。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0: 成功；非0: 失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
scan_code_freeze	否	ScanCodeFreezeResponse	authen_info 存在时必填。详见 ScanCodeFreezeResponse。

ScanCodeFreezeResponse 结构

参数名	必填	类型	说明
code_url	是	String	二维码 url。
nonce_str	是	String(32)	随机字符串。

示例

● 输入参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "83F7325E12B8CC780EC907230C7CBC7EDC592C907434C3DA83DB9AE3822EAB0C"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 2,
      "sub_pay_platform": 200,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "freeze_content": {
      "body": "body",
      "out_freeze_no": "sz010002cz11566388969",
      "freeze_fee": 10,
      "fee_type": "CNY"
    },
    "order_client": {
      "staff_id": "1206",
      "machine_no": "00-05-9a-3c-7a-00",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "824",
      "spbill_create_ip": "90.0.00.0"
    }
  }"
}
```

● 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 3073620540,
    "internal_status": 0,
    "scan_code_freeze": {
      "code_url": "https://\mobilecodec.alipay.com/show.htm?code=aax05690yvunvarg8pacbc2",
      "nonce_str": "IFLLIzq3n8qvYBcuxKqFRL054gsQIY2"
    }
  }",
}
```

```
"authn_info": {  
  "a": {  
    "authn_type": 1,  
    "authn_code": "2686CE142CAA36EF8E5EDAD0E672745B15701A617C4BBF2D69823A721A94C72D"  
  }  
}
```

押金消费

最近更新时间：2023-08-17 10:35:12

接口描述

项	值
url	https://pay.qcloud.com/cpay/deposit_consume

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	三级参数名	必填	类型	说明
pay_content	out_trade_no	-	是	String(32)	由客户端生成的订单号，前缀必须是云支付订单前缀。
	total_fee	-	是	Number(32)	订单总金额，单位：分。
	fee_type	-	是	String	货币类型（目前只支持人民币，请填写 CNY）。
	body	-	是	String(128)	商品或订单简要描述。 商品描述交易字段格式根据不同的应用场景按照以下格式： ● PC 网站：传入浏览器打开的网站主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 公众号：传入公众号名称-实际商品名称，例如：腾讯形象店-image-QQ 公仔。 ● H5：应用在浏览器网页上的场景，传入浏览器打开的移动网页的主页 title 名-实际商品名称，例如：腾讯充值中心-QQ 会员充值。 ● 线下门店：门店品牌名-城市分店名-实际商品名称，例如：image 形象店-深圳腾大-QQ 公仔。 ● App：需传入应用市场上的 App 名字-实际商品名称，天天爱消除-游戏充值。
	detail	-	否	String	商品详细列表，由 json 转化而来，json 结构见公共数据结构 Detail。
	remark	-	否	String(127)	支付备注信息。
	wxpay_pay_content_ext*	attach	否	String(127)	附加数据，记录子商户自定义数据。
		goods_tag	否	String(32)	商品标记，代金券或立减优惠功能的参数。
		product_id	否	String(32)	商品 ID，子商户自定义。

		limit_pay	否	String(32)	定支付方式，目前只能是 no_credit，指定不能使用信用卡支付。
	out_freeze_no	-	是	String(32)	冻结押金授权单号。
	is_confirm_unfreeze	-	是	Bool	是否解冻用户金额（微信支付目前只能为 true，支付宝可以为 true/false）。
pay_mch_key	pay_platform	-	是	Number(32)	第三方支付类型，详见枚举 PayPlatform。
	out_mch_id	-	是	String(32)	服务商对外账号。
	out_sub_mch_id	-	是	String(32)	子商户对外账号。
	out_shop_id	-	是	String(32)	门店对外账号。
order_client	device_id	-	是	String(64)	子商户自定义，终端设备号。
	staff_id	-	否	String(64)	子商户自定义，店员 ID。
	terminal_type	-	是	Number(32)	终端类型，1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	sdk_version	-	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	-	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	-	否	String(127)	设备 sn 号。
nonce_str	-	-	是	String(32)	随机字符串。

*: 微信支付的订单才需要 wxpay_pay_content_ext。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0: 成功; 非0: 失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
deposit_consume	否	DepositConsumeResponse	authen_info 存在时必填。详见 DepositConsumeResponse。

DepositConsumeResponse 结构

参数名	必填	类型	说明
order_content	是	OrderContent	订单信息，详见公共数据结构 OrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。

nonce_str	是	String(32)	随机字符串。
-----------	---	------------	--------

示例

● 输入参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "88AB365DFF57E383FDF5D189A3177274A0054C38BAAC8549D45E57013E7429EE"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2I4B"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "order_client": {
      "staff_id": "1206",
      "machine_no": "00-05-9a-3c-7a-00",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "824",
      "spbill_create_ip": "90.0.00.0"
    },
    "pay_content": {
      "body": "body",
      "out_freeze_no": "sz010002cz11566541562",
      "is_confirm_unfreeze": true,
      "out_trade_no": "sz010002cz11566541614",
      "total_fee": 1,
      "fee_type": "CNY"
    }
  }"
}
```

● 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 1915124595,
    "internal_status": 0,
    "deposit_consume": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01IXKA6DKGjNzr2I4B",
        "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
        "out_shop_id": "sz011biKxOguirmBqiFR",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx998fa95558590e1c",
          "sub_app_id": ""
        }
      }
    }
  }"
```

```
    },
    "order_content": {
      "out_trade_no": "sz010002cz11566541614",
      "transaction_id": "4200000377201908231038860810",
      "trade_type": 1,
      "time_expire": 1566541734,
      "time_end": 1566541615,
      "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
      "create_time": 1566541614,
      "last_update_time": 1566541615,
      "is_transforming": false,
      "caller_notify_url": "",
      "total_fee": 1,
      "fee_type": "CNY",
      "cash_fee": 1,
      "cash_fee_type": "CNY",
      "settlement_total_fee": 1,
      "body": "body",
      "wxpay_order_content_ext": {
        "current_trade_state": 14,
        "attach": "",
        "bank_type": "CFT",
        "goods_tag": "X293123",
        "limit_pay": ""
      },
      "out_freeze_no": "sz010002cz11566541562",
      "freeze_fee": 10,
      "is_deposit_mode": true,
      "is_confirm_unfreeze": true,
      "manual_unfreeze_fee": 9,
      "freeze_id": "4200000377201908231038860810"
    },
    "nonce_str": "3cRILhFBSAK4wsTpy1eHmgJlSThLncdV"
  }
},
{
  "a": {
    "authen_type": 1,
    "authen_code": "0F8B6390E46E99D0967F2F15C48D9FB0527A63C9BD3E2D341A3BFECA3EEC7B0A"
  }
}
}
```

押金解冻

最近更新时间：2023-08-17 10:35:12

接口描述

项	值
url	https://pay.qcloud.com/cpay/unfreeze

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
unfreeze_content	out_freeze_no	是	String(32)	冻结单号。
	out_unfreeze_no	是	String(32)	解冻单号。
	unfreeze_fee	是	Number(32)	解冻金额，金额分。
	remark	否	String(128)	解冻备注信息。
pay_mch_key	pay_platform	是	Number(32)	第三方支付类型。
	out_mch_id	否	String(32)	服务商对外账号。
	out_sub_mch_id	否	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
	is_sub_mch_admin	是	Bool	操作者是否为商户管理员（此参数是为了校验解冻人身份，操作解冻人必须拥有退款权限。true 代表操作者是商户管理员，该角色默认拥有退款权限。当为 false 时与 authorization_staff_id 配合使用校验退款权限）。
	authorization_staff_id	否	String(64)	如果 is_sub_mch_admin 为 false，则此项为必填，如果为 true 则此项不填。 1. 如果当前 staff_id 有退款权限则传自身的 staff_id。 2. 如果是其他人授权则填授权者的 staff_id。 3. 机具调用传机具内配置的 staff_id。
order_client	device_id	是	String(64)	子商户自定义，终端设备号。
	terminal_type	是	Number(32)	终端类型，1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。

	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	否	String(127)	设备 sn 号。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
unfreeze	否	UnfreezeResponse	authen_info 存在时必填。详见 UnfreezeResponse。

UnfreezeResponse 结构

参数名	必填	类型	说明
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
unfreeze_order_content	否	UnfreezeOrderContent	押金解冻单信息，详见公共数据结构 UnfreezeOrderContent。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "7CFA24835EFC1E4349D45BC9E0B821196B5D56F1CDDFB1ED5F23951B159554D5"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "is_sub_mch_admin": false,
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "sub_pay_platform": 100,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "authorization_staff_id": "zz123",
      "pay_platform": 1,
      "out_mch_id": "sz01IXKA6DKGjNzr2I4B"
    }
  }
```

```
    },
    "unfreeze_content": {
      "out_unfreeze_no": "sz010002cz11566551626",
      "unfreeze_fee": 10,
      "remark": "test",
      "out_freeze_no": "sz010002cz11566551329"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "order_client": {
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "824",
      "spbill_create_ip": "90.0.00.0"
    }
  }
}
```

● 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 1733154361,
    "internal_status": 0,
    "unfreeze": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
        "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
        "out_shop_id": "sz011biKxOguirmBqiFR",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx998fa95558590e1c",
          "sub_app_id": ""
        }
      }
    },
    "unfreeze_order_content": {
      "out_freeze_no": "sz010002cz11566551329",
      "out_unfreeze_no": "sz010002cz11566551626",
      "trade_type": 1,
      "nonce_str": "oyuP18zF30egrgKE",
      "create_time": 1566551626,
      "last_update_time": 1566551627,
      "is_transforming": false,
      "unfreeze_fee": 10,
      "remark": "test",
      "freeze_id": "4200000384201908236723637327",
      "fee_type": "CNY",
      "wxpay_unfreeze_order_content_ext": {
        "current_trade_state": 2
      }
    },
    "nonce_str": "EM8hO8MOMoPsbRXVElOOcdmgZx5WZ9FZ"
  }",
  "authen_info": {
    "a": {
      "authen_type": 1,

```

```
"authen_code": "459BFAD5F0524252F1BBBE12B005ED8C0BB0D97B146B50E3A953A62D677E5AF5"  
}  
}  
}
```

查询押金冻结单

最近更新时间：2023-08-17 10:35:12

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_freeze_order

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	pay_platform	是	Number(32)	第三方支付类型。
	out_mch_id	是	String(32)	服务商对外账号。
	out_sub_mch_id	是	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
order_client	device_id	否	String(64)	子商户自定义，终端设备号。
	staff_id	否	String(64)	子商户自定义，店员 ID。
	terminal_type	是	Number(32)	终端类型，1: Windows，2: Andriod，3: iOS，4: Linux，100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	否	String(127)	设备 sn 号。
out_freeze_no	-	是	String(32)	冻结押金授权单号。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。

authen_info	否	AuthenInfo	认证信息，详见接口调用说明。
-------------	---	------------	----------------

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_freeze_order	否	QueryFreezeOrderResponse	authen_info 存在时必填。详见 QueryFreezeOrderResponse。

QueryFreezeOrderResponse 结构

参数名	必填	类型	说明
freeze_order_content	是	FreezeOrderContent	冻结单信息，详见公共数据结构 FreezeOrderContent。
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
order_client	否	OrderClient	客户端信息列表，详见公共数据结构 OrderClient。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "9158B1D249EE83F8A406F07056C5C38094BE6569B64E2D2E70128AA7511CA2BB"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B"
    },
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "out_freeze_no": "sz010002cz11564384955",
    "order_client": {
      "staff_id": "1192",
      "machine_no": "1111",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "12345",
      "spbill_create_ip": "90.0.00.0"
    }
  }"
}
```

● 输出参数示例:

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "3B11C7489804E267C917DBA8508D403A3CF53C895272AC43587A631325629A18"
    }
  },
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 1971094428,
    "internal_status": 0,
    "query_freeze_order": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
        "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
        "out_shop_id": "sz011biKxOguirmBqiFR",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx998fa95558590e1c",
          "open_id": "ovJrDwsjamQql-k7Zr1Q-EaKFQ0k"
        }
      }
    },
    "freeze_order_content": {
      "out_freeze_no": "sz010002cz11564386781",
      "trade_type": 1,
      "author_code": "134550264496882447",
      "time_expire": 1564386901,
      "time_end": 1564386782,
      "create_time": 1564386781,
      "last_update_time": 1564386783,
      "is_transforming": false,
      "freeze_fee": 10,
      "fee_type": "CNY",
      "cash_fee": 10,
      "cash_fee_type": "CNY",
      "settlement_total_fee": 10,
      "body": "body",
      "wxpay_freeze_order_content_ext": {
        "current_trade_state": 2,
        "bank_type": "CFT",
        "goods_tag": "X293123"
      }
    },
    "nonce_str": "hFiSO4JaMpDB38qWELxmG45Maue5Wdzp"
  }"
}
```

查询押金解冻单

最近更新时间：2023-08-17 10:35:12

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_unfreeze_order

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

一级参数名	二级参数名	必填	类型	说明
pay_mch_key	pay_platform	是	Number(32)	第三方支付类型。
	out_mch_id	是	String(32)	服务商对外账号。
	out_sub_mch_id	是	String(32)	子商户对外账号。
	out_shop_id	是	String(32)	门店对外账号。
order_client	device_id	否	String(64)	子商户自定义，终端设备号。
	staff_id	否	String(64)	子商户自定义，店员 ID。
	terminal_type	是	Number(32)	终端类型，1: Windows, 2: Andriod, 3: iOS, 4: Linux, 100: 其他。
	sdk_version	是	String(10)	云支付 SDK 版本号。
	spbill_create_ip	是	String(16)	调用云支付 API 的机器 IP。
	sn_code	否	String(127)	设备 sn 号。
out_freeze_no	-	是	String(32)	冻结押金授权单号。
out_unfreeze_no	-	否	String(32)	押金解冻单号。
nonce_str	-	是	String(32)	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。

authen_info	否	AuthenInfo	认证信息，详见接口调用说明。
-------------	---	------------	----------------

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_unfreeze_order	否	QueryUnfreezeOrderResponse	authen_info 存在时必填。详见 QueryUnfreezeOrderResponse。

QueryUnfreezeOrderResponse 结构

参数名	必填	类型	说明
pay_mch_key	是	PayMchKey	支付商户信息，详见公共数据结构 PayMchKey。
unfreeze_order_content	否	UnfreezeOrderContent[]	押金解冻单信息，详见公共数据结构 UnfreezeOrderContent。
order_client	否	OrderClient[]	客户端信息列表，详见公共数据结构 OrderClient。
nonce_str	是	String(32)	随机字符串。

示例

- 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "9158B1D249EE83F8A406F07056C5C38094BE6569B64E2D2E70128AA7511CA2BB"
    }
  },
  "request_content": "{
    "pay_mch_key": {
      "pay_platform": 1,
      "out_shop_id": "sz011biKxOguirmBqiFR",
      "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
      "out_mch_id": "sz01IXKA6DKGjNzr2l4B"
    },
    "out_unfreeze_no": "sz010002cz11566551626",
    "nonce_str": "E94C00688C3F429CA2B0B396BF823548",
    "out_freeze_no": "sz010002cz11566551329",
    "order_client": {
      "terminal_type": 1,
      "sdk_version": "1.0",
      "device_id": "12345",
      "spbill_create_ip": "90.0.00.0"
    }
  }"
}
```

● 输出参数示例:

```
{
  "response_content": "{
    "status": 0,
    "description": "\u64CD\u4F5C\u6210\u529F\u3002",
    "log_id": 3804805851,
    "internal_status": 0,
    "query_unfreeze_order": {
      "pay_mch_key": {
        "pay_platform": 1,
        "out_mch_id": "sz01IXKA6DKGjNzr2l4B",
        "out_sub_mch_id": "sz01KzuCUOmw8yjtPite",
        "out_shop_id": "sz011biKxOguirmBqiFR",
        "sub_pay_platform": 100,
        "wxpay_pay_mch_key_ext": {
          "app_id": "wx998fa95558590e1c",
          "sub_app_id": ""
        }
      }
    },
    "unfreeze_order_content": [{
      "out_freeze_no": "sz010002cz11566551329",
      "out_unfreeze_no": "sz010002cz11566551626",
      "unfreeze_id": "",
      "trade_type": 1,
      "time_end": 0,
      "nonce_str": "oyuP18zF30egrgKE",
      "create_time": 1566551626,
      "last_update_time": 1566551627,
      "is_transforming": false,
      "unfreeze_fee": 10,
      "remark": "test",
      "freeze_id": "4200000384201908236723637327",
      "fee_type": "CNY",
      "trade_state": 4,
      "wxpay_unfreeze_order_content_ext": {
        "current_trade_state": 2
      }
    }],
    "nonce_str": "WED5XRwbEO9QgAlu2mdt4Fo5szYoutXs",
    "order_client": [{
      "device_id": "824",
      "machine_no": "1111",
      "terminal_type": 1,
      "sdk_version": "1.0",
      "spbill_create_ip": "90.0.00.0",
      "sub_terminal_type": 0,
      "sn_code": ""
    }]
  }",
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "9E0C43F215A270DE8E53EA4505C503DFD6B73513DB6C9934E434660FF4508380"
    }
  }
}
```

多条件查询订单信息

最近更新时间：2020-06-29 16:28:50

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_order_list

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

参数名	必填	类型	说明
sub_pay_platforms	否	Number(32)[]	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
out_sub_mch_id	是	String	云支付分配的商户 ID。
out_shop_id	否	String	云支付分配的门店 ID。
staff_id	否	String	员工 ID，如果设置，则必须同时设置 out_shop_id。
device_id	否	String	设备 ID，如果设置，则必须同时设置 out_shop_id。
qrcode_id	否	String	二维码 ID，如果设置，则必须同时设置 out_shop_id。
source_type	否	Number(32)	订单来源，1：一码付，2：店员户外二维码收银，3：绑定二维码，4：定额二维码，5：会员卡充值，6：临时二维码，7：会员卡一码付。
order_type	是	Number(32)	单据类型，1：支付订单，2：退款单，3：全部单据，4：冻结单，5：解冻单。
start_time	是	Number(64)	查询开始时间。
end_time	是	Number(64)	查询结束时间。
page_num	是	Number(32)	页码。
page_size	是	Number(32)	单页条数。
trade_type	否	Number(32)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
pay_time_type	否	Number(32)	支付时间类型，1：支付单创建时间，2：支付单成功时间，默认为2。
refund_time_type	否	Number(32)	退款时间类型，1：退款创建时间，2：退款发起时间，默认为2。
time_filter	否	Number(32)	时间过滤字段，1：创建时间，2：更新时间，默认为2。

status	否	Number(32)	查出处于该状态的单，0：只查询成功状态的单，1：查询所有状态的单，默认为0。
nonce_str	是	String	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败或者需要重试，具体见实际返回的错误码。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_order_list	否	QueryOrderListResponse	authen_info 存在时必填。详见 QueryOrderListResponse。

QueryOrderListResponse 结构

参数名	必填	类型	说明
nonce_str	是	String(32)	随机字符串。
total_count	否	Number(32)	汇总数。
order_details	否	OrderDetail[]	交易明细结构体。

OrderDetail 结构

参数名	必填	类型	说明
order[1]	否	Order	订单结构体，见本节 Order。
refund_order[1]	否	RefundOrder	退款单结构体，见本节 RefundOrder。
freeze_order[1]	否	FreezeOrder	冻结单结构体，见本节 FreezeOrder。
unfreeze_order[1]	否	UnfreezeOrder	解冻单结构体，见本节 UnfreezeOrder。

[1]: order、refund_order、freeze_order 和 unfreeze_order 只能包含一个。

Order 结构

参数名	必填	类型	说明
out_mch_id	否	String	云支付分配给服务商的 ID。
out_sub_mch_id	否	String	云支付分配给子商户的 ID。

shop_name	否	String	门店名称。
staff_id	否	String	店员 ID。
staff_name	否	String	店员姓名。
device_id	否	String	设备 ID。
device_name	否	String	设备名称。
qrcode_id	否	String	二维码 ID。
qrcode_name	否	String	二维码名称。
source_type	否	Number(3 2)	订单来源，1：一码付，2：店员户外二维码收银，3：绑定二维码，4：定额二维码，5：会员卡充值，6：临时二维码，7：会员卡一码付。
out_card_id	否	String	会员卡 ID。
sub_pay_platform	否	Number(3 2)	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
trade_type	否	Number(3 2)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
out_trade_no	是	String	子商户订单号，云支付系统单 Set 内全局唯一。
transaction_id	否	String	第三方支付平台的订单号。
total_fee	否	Number(6 4)	订单总金额。
fee_type	否	String	货币类型。
cash_fee	否	Number(6 4)	现金支付金额。
cash_fee_type	否	String	现金支付货币类型。
settlement_total_fee	否	Number(6 4)	应结支付金额。
discount_fee	否	Number(6 4)	优惠金额。
recharge_bonus_fee	否	Number(6 4)	会员卡充值赠送金额。
is_card_recharge	否	Bool	是否会员卡充值单。
poundage	否	Number(6 4)	第三方账单给出的手续费。
coupon_infos	否	CouponInfo	优惠信息，见公共结构体 CouponInfo。
remaining_settlement_fee	否	Number(6 4)	订单实收金额。
income_fee	否	Number(6 4)	入账金额。
time_end	否	Number(6 4)	支付完成时间。

create_time	是	Number(64)	订单创建时间。
last_update_time	是	Number(64)	订单最近更新时间。
body	否	String	商品或订单简要描述，下单时必须。
detail	否	String	商品详细列表。
client_flow_id	否	String	客户端流水单号。
remark	否	String	支付备注信息。
refunded_fee	否	Number(64)	已退金额。
refund_create_time	否	Number(64)	最近退款创建时间。
refund_time	否	Number(64)	最近退款成功时间。
refunded_settlement_fee	否	Number(64)	已退结算金额。
wxpay_current_trade_state	否	Number(32)	微信订单状态，见枚举值定义 WxpayOrderState。
attach	否	String	附加数据，记录子商户自定义数据。
bank_type	否	String	刷卡支付时特有，付款银行类型，遵守字符型银行编码规范。
goods_tag	否	String	商品标记，代金券或立减优惠功能的参数。
scene_info	否	String	支付的场景信息 json 串。
alipay_current_trade_state	否	Number(32)	支付宝订单状态，见枚举值定义 AlipayOrderState。
discountable_amount	否	Number(64)	参与优惠的金额。
undiscountable_amount	否	Number(64)	不参与优惠的金额。
membership_number	否	String	会员卡号。
uid	否	Number(64)	会员 uid。
credits	否	Number(64)	实际消费的积分数量。
record_current_trade_state	否	Number(32)	记账订单状态，见枚举值定义 RecordOrderState。
out_freeze_no	否	String	预授权号。
freeze_fee	否	Number(64)	冻结金额。
is_deposit_mode	否	Bool	押金下模式。
is_confirm_unfreeze	否	Bool	是否在消费并且解冻用户金额。

manual_unfreeze_fee	否	Number(64)	消费且解冻金额。
freeze_id	否	String	冻结单 freeze_id。

RefundOrder 结构

参数名	必填	类型	说明
out_mch_id	否	String	云支付分配给服务商的 ID。
out_sub_mch_id	否	String	云支付分配给子商户的 ID。
out_shop_id	否	String	云支付分配给门店的 ID。
shop_name	否	String	门店名称。
staff_id	否	String	店员 ID。
staff_name	否	String	店员姓名。
device_id	否	String	设备 ID。
device_name	否	String	设备名称。
qrcode_id	否	String	二维码 ID。
qrcode_name	否	String	二维码名称。
source_type	否	Number(32)	订单来源，1：一码付，2：店员户外二维码收银，3：绑定二维码，4：定额二维码，5：会员卡充值，6：临时二维码，7：会员卡一码付。
out_card_id	否	String	会员卡 ID。
sub_pay_platform	否	Number(32)	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
trade_type	否	Number(32)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
out_trade_no	是	String	子商户订单号，云支付系统单 Set 内全局唯一。
total_fee	是	Number(64)	订单总金额。
out_refund_no	是	String	子商户退款单号，云支付系统单 Set 内全局唯一。
refund_id	否	String	第三方支付平台的退款单号。
refund_fee	是	Number(64)	本次退款总金额。
settlement_refund_fee	否	Number(64)	本次退款结算金额。
refund_fee_type	是	String	本次退款总金额货币类型。
poundage	否	Number(64)	第三方账单给出的手续费。
coupon_infos	否	CouponInfo	退款优惠信息，见公共结构体 CouponInfo。
income_fee	否	Number(64)	入账金额。

		4)	
create_time	是	Number(64)	退款单创建时间。
last_update_time	是	Number(64)	退款单最近更新时间。
refund_start_processing_time	否	Number(64)	退款发起时间。
pay_last_update_time	否	Number(64)	支付成功时间。
refund_reason	否	String	退款原因。
client_refund_flow_id	否	String	客户端退款流水单号。
wxpay_refund_state	否	Number(32)	微信退款状态，见枚举值定义 WxpayRefundOrderState。
cash_refund_fee	否	Number(64)	现金退款金额。
cash_refund_fee_type	否	String	现金退款货币类型。
alipay_refund_state	否	Number(32)	支付宝退款状态，见枚举值定义 AlipayRefundOrderState。
credits	否	Number(64)	实际退款的积分数量。
record_refund_state	否	Number(32)	记账退款状态，见枚举值定义 RecordRefundOrderState。
is_deposit_mode	否	bool	押金下模式。
out_freeze_no	否	String	外部授权单号（押金消费退款必填）。
freeze_fee	否	Number(64)	冻结金额。

FreezeOrder 结构

参数名	必填	类型	说明
out_mch_id	否	String	云支付分配给服务商的 ID。
out_sub_mch_id	否	String	云支付分配给子商户的 ID。
out_shop_id	否	String	云支付分配给门店的 ID。
shop_name	否	String	门店名称。
staff_id	否	String	店员 ID。
staff_name	否	String	店员姓名。
device_id	否	String	设备 ID。
device_name	否	String	设备名称。
sub_pay_platform	否	Number(32)	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。

trade_type	否	Number(32)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
out_freeze_no	是	String	授权订单号。
freeze_id	否	String	第三方授权单号。
freeze_fee	是	Number(64)	冻结金额。
fee_type	是	String	货币类型。
cash_fee	否	Number(64)	现金支付金额。
cash_fee_type	否	String	现金支付货币类型。
settlement_total_fee	否	Number(64)	应结支付金额。
discount_fee	否	Number(64)	针对商户来说的优惠金额：total_fee - settlement_total_fee。
total_consume_fee	否	Number(64)	总消费金额。
total_manual_unfreeze_fee	否	Number(64)	总解冻金额。
total_auto_unfreeze_fee	否	Number(64)	自动30天后自动的解冻金额。
total_refund_fee	否	Number(64)	总退款金额。
coupon_infos	否	CouponInfo	优惠信息（第三方平台返回），见公共结构体 CouponInfo。
create_time	否	Number(64)	订单创建时间。
last_update_time	否	Number(64)	订单最近更新时间。
time_end	否	Number(64)	支付完成时间（第三方返回）。
body	否	String	商品或订单简要描述，外部必填参数。
detail	否	String	优惠信息。
exchange_rate	否	Number(64)	汇率。
remark	否	String	支付备注信息。
wxpay_current_trade_state	否	Number(32)	微信冻结单状态，见枚举值定义 WxpayFreezeOrderState。
alipay_current_trade_state	否	Number(32)	支付宝冻结单状态，见枚举值定义 AlipayFreezeOrderState。

UnfreezeOrder 结构

--	--	--

参数名	必填	类型	说明
out_mch_id	否	String	云支付分配给服务商的 ID。
out_sub_mch_id	否	String	云支付分配给子商户的 ID。
out_shop_id	否	String	云支付分配给门店的 ID。
shop_name	否	String	门店名称。
staff_id	否	String	店员 ID。
staff_name	否	String	店员姓名。
device_id	否	String	设备 ID。
device_name	否	String	设备名称。
sub_pay_platform	否	Number(32)	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
trade_type	否	Number(32)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
out_freeze_no	否	String	授权订单号。
out_unfreeze_no	是	String	解冻单号。
unfreeze_id	否	String	第三方操作单号。
freeze_id	否	String	对应冻结单中 freeze_id。
unfreeze_fee	否	Number(64)	解冻金额。
fee_type	否	String	货币类型。
time_end	否	Number(64)	解冻完成时间。
create_time	是	Number(64)	订单创建时间。
last_update_time	是	Number(64)	订单最近更新的时间。
remark	否	String	备注信息。
wxpay_current_trade_state	否	Number(32)	微信解冻单状态，见枚举值定义 WxpayUnfreezeOrderState。
alipay_current_trade_state	否	Number(32)	支付宝解冻单状态，见枚举值定义 AlipayUnfreezeOrderState。

示例

• 输入参数示例：

```
{
  "request_content": "{
    "sub_pay_platforms": [],
    "order_type": 3,
    "start_time": 1567008000,
    "end_time": 1567094399,
```

- **输出参数示例:**

版权所有：腾讯云计算（北京）有限责任公司

[illegible]

子商户交易汇总查询

最近更新时间：2020-06-29 16:40:43

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_order_list_overview

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

参数名	必填	类型	说明
sub_pay_platforms	否	Number(32) []	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
out_sub_mch_id	是	String	云支付分配的商户 ID。
out_shop_id	否	String	云支付分配的门店 ID。
staff_id	否	String	员工 ID。
device_id	否	String	设备 ID。
qrcode_id	否	String	二维码 ID。
source_type	否	Number(32))	订单来源，1：一码付，2：店员户外二维码收银，3：绑定二维码，4：定额二维码，5：会员卡充值，6：临时二维码，7：会员卡一码付。
order_type	否	Number(32))	单据类型，1：支付订单，2：退款单，3：全部单据，4：冻结单，5：解冻单。
start_time	是	Number(64))	查询开始时间。
end_time	是	Number(64))	查询结束时间。
trade_type	否	Number(32))	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
out_trade_no	否	String	订单号。
out_refund_no	否	String	退款单号。
pay_time_type	否	Number(32))	支付时间类型，1：支付单创建时间，2：支付单成功时间，默认为2。
refund_time_type	否	Number(32))	退款时间类型，1：退款创建时间，2：退款发起时间，默认为2。
time_filter	否	Number(32))	时间过滤字段，1：创建时间，2：更新时间，默认为2。

status	否	Number(32)	查出处于该状态的单，0：只查询成功状态的单，1：查询所有状态的单，默认为0。
nonce_str	是	String	随机字符串。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败或者需要重试，具体见实际返回的错误码。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
Internal_status	是	Number(32)	调试使用，调用者可以不予理会。
query_order_list_overview	否	QueryOrderListOverviewResponse	authen_info 存在时必填。详见 QueryOrderListOverviewResponse。

QueryOrderListOverviewResponse 结构

参数名	必填	类型	说明
nonce_str	是	String(32)	随机字符串。
total_count	否	Number(32)	汇总数。
overviews	否	OrderStatClientInfo[]	交易汇总结构体。

OrderStatClientInfo 结构

参数名	必填	类型	说明
out_sub_mch_id	否	String	云支付分配的商户 ID。
out_shop_id	否	String	云支付分配的门店 ID。
staff_id	否	String	员工 ID。
device_id	否	String	设备 ID。
qrcode_id	否	String	二维码 ID。
source_type	否	Number(32)	订单来源，1：一码付，2：店员户外二维码收银，3：绑定二维码，4：定额二维码，5：会员卡充值，6：临时二维码，7：会员卡一码付。
sub_pay_platform	否	Number(32)	子支付平台列表，100：普通微信支付，200：普通支付宝，300：代金券/校园卡，401：银行卡记账，402：现金记账。
trade_type	否	Number(32)	交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。

fee_type	是	String	货币类型。
success_count	是	Number(64)	(交易量) 不含撤单的, 扣款成功交易笔数, 有可能是负值。
success_amount	是	Number(64)	不含撤单的, 扣款成功交易金额, 有可能是负值。
settle_amount	是	Number(64)	操作人扣款成功结算金额 - 操作人退款结算金额 - 操作人扣款成功撤单结算金额。
discount_amount	是	Number(64)	net_amount - settle_amount。
pay_settle_amount	是	Number(64)	扣款成功结算金额。
refund_settle_amount	是	Number(64)	退款结算金额。
refund_create_count	是	Number(64)	退款发起笔数。
refund_create_amount	是	Number(64)	退款发起总额。
poundage	是	Number(64)	扣款成功手续费 - 撤单返还手续费 - 退款成功返还的手续费, 有可能是负值。
income_amount	是	Number(64)	入账金额。
order_refunded_amount	是	Number(64)	订单已退金额。
sub_mch_recharge_coupon_count	否	Number(64)	必返回, 商户充值优惠券数量。
sub_mch_recharge_coupon_amount	否	Number(64)	必返回, 商户充值优惠券金额。
platform_recharge_coupon_count	否	Number(64)	必返回, 平台充值优惠券数量。
platform_recharge_coupon_amount	否	Number(64)	必返回, 平台充值优惠券金额。
others_recharge_coupon_count	否	Number(64)	必返回, 其他充值优惠券数量。
others_recharge_coupon_amount	否	Number(64)	必返回, 其他充值优惠券金额。
sub_mch_non_recharge_coupon_count	否	Number(64)	必返回, 商户免充值优惠券数量。
sub_mch_non_recharge_coupon_amount	否	Number(64)	必返回, 商户免充值优惠券金额。
platform_non_recharge_coupon_count	否	Number(64)	必返回, 平台免充值优惠券数量。
platform_non_recharge_coupon_amount	否	Number(64)	必返回, 平台免充值优惠券金额。

others_non_recharge_coupon_count	否	Number(64)	必返回，其他免充值优惠券数量。
others_non_recharge_coupon_amount	否	Number(64)	必返回，其他免充值优惠券金额。
freeze_count	否	Number(64)	必返回，冻结笔数。
freeze_amount	否	Number(64)	必返回，冻结金额。
direct_unfreeze_count	否	Number(64)	必返回，直接（调解冻接口）解冻笔数。
direct_unfreeze_amount	否	Number(64)	必返回，直接（调解冻接口）解冻金额。
consumed_unfreeze_amount	否	Number(64)	必返回，消费完成后自动解冻金额。
auto_unfreeze_amount	否	Number(64)	必返回，冻结单创建成功30天后第三方支付平台自动解冻金额。

示例

● 输入参数示例：

```
{
  "request_content": "{
    "sub_pay_platforms": [100],
    "out_sub_mch_id": "sz01F1Nq56FdmWj5vx25",
    "out_shop_id": "sz01t2emx246y307BVon",
    "staff_id": "178424",
    "device_id": "61204",
    "source_type": 5,
    "order_type": 1,
    "start_time": 1567078099,
    "end_time": 1567083622,
    "trade_type": 1,
    "pay_time_type": 1,
    "refund_time_type": 1,
    "time_filter": 1
  }",
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "E05DB4F0E48798CF74369602EB090D0B1640E75F29BF375AB81BA0E6CF73E980"
    }
  }
}
```

● 输出参数示例：

```
{
  "response_content": "{
    "status": 0,
    "description": "u64CDu4F5Cu6210u529Fu3002",
    "log_id": 761035717,
    "Internal_status": 0,
  }
```

```
"query_order_list_overview":{
  "overviews":[
    {
      "fee_type":"CNY",
      "success_count":415,
      "success_amount":743380,
      "settle_amount":743380,
      "discount_amount":0,
      "pay_settle_amount":743380,
      "refund_settle_amount":0,
      "refund_create_count":0,
      "refund_create_amount":0,
      "poundage":2341,
      "income_amount":741039,
      "order_refunded_amount":0,
      "sub_mch_recharge_coupon_count":3,
      "sub_mch_recharge_coupon_amount":20,
      "platform_recharge_coupon_count":0,
      "platform_recharge_coupon_amount":0,
      "others_recharge_coupon_count":0,
      "others_recharge_coupon_amount":0,
      "sub_mch_non_recharge_coupon_count":0,
      "sub_mch_non_recharge_coupon_amount":0,
      "platform_non_recharge_coupon_count":0,
      "platform_non_recharge_coupon_amount":0,
      "others_non_recharge_coupon_count":0,
      "others_non_recharge_coupon_amount":0,
      "freeze_count":0,
      "freeze_amount":0,
      "direct_unfreeze_count":0,
      "direct_unfreeze_amount":0,
      "consumed_unfreeze_amount":0,
      "auto_unfreeze_amount":0
    }
  ],
  "nonce_str":"KdGI3KZabkasaTQITNsSIyvFI7S86No4"
},
"authen_info":{
  "a":{
    "authen_type":1,
    "authen_code":"66FCA9C535D2E166FC103731859BA7C9298A580BCD8A87F9869461689A39F4AA"
  }
}
}
```

门店接口

查询门店信息

最近更新时间：2023-05-26 11:44:26

接口描述

项	值
url	https://pay.qcloud.com/cpay/query_sub_mch_shop_info

输入参数

参数名	必填	类型	说明
request_content	是	String	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	认证信息，详见接口调用说明。

RequestContent 结构

参数名	必填	类型	说明
out_mch_id	是	String(32)	云支付分配给服务商的账号。
out_sub_mch_id	是	String(32)	云支付分配给子商户的账号。
nonce_str	是	String(32)	随机字符串。
page_num	是	Number(32)	页码（从1开始）。
page_size	是	Number(32)	单页条数。

输出参数

参数名	必填	类型	说明
response_content	是	ResponseContent	请求内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	认证信息，详见接口调用说明。

ResponseContent 结构

参数名	必填	类型	说明
status	是	Status	错误码，详见 Status。0：成功；非0：失败或者需要重试，具体见实际返回的错误码。
description	否	String(255)	错误描述。
log_id	是	Number(32)	消息 ID。
internal_statuses	是	Number(32)	调试使用，调用者可以不予理会。
query_shop_info	否	QueryShopInfoResponse	authen_info 存在时必填。详见 QueryShopInfoResponse。

QueryShopInfoResponse 结构

参数名	必填	类型	说明
nonce_str	是	String(32)	随机字符串。
shop_infos	否	ShopInfo []	门店信息，详见公共数据结构 ShopInfo。 status 为0时返回。
total_count	是	Number(32)	数据总数。 status 为0时返回。

示例

● 输入参数示例：

```
{
  "authen_info": {
    "a": {
      "authen_type": 1,
      "authen_code": "B2BEE6784FDA86712FD1EA4824B734F6CA074F780D51E1D6EAE53BB423A05E23"
    }
  },
  "request_content": "{
    \"out_mch_id\": \"sz01PWfGyu80piEk3ENs\",
    \"out_sub_mch_id\": \"sz01MKDW9Ac06YnPkCnf\",
    \"nonce_str\": \"7FD9E45968E64ADD9BFE71D0DBFBEF96\",
    \"page_num\": 1,
    \"page_size\": 50
  }"
}
```

● 输出参数示例：

```
{
  "response_content": "{
    \"status\":0,
    \"description\":\"\\u64CD\\u4F5C\\u6210\\u529F\\u3002\",
    \"log_id\":3976286590,
    \"internal_status\":0,
    \"query_shop_info\":{
      \"shop_infos\":[
        {
          \"shop_id\":\"229776\",
          \"shop_name\":\"\\u8D27\\u67B6\\u95E8\\u5E97\",
          \"branch_name\":\"\",
          \"province\":\"\\u5E7F\\u4E1C\\u7701\",
          \"city\":\"\\u6DF1\\u5733\\u5E02\",
          \"district\":\"\\u5357\\u5C71\\u533A\",
          \"address\":\"\\u0078\\u0078\\u8DEF\\u0078\\u0078\\u53F7\\u0078\\u0078\\u680B\\u0078\\u0078\\u623F\",
          \"coordinate_type\":0,
          \"longitude\":\"\",
          \"latitude\":\"\",
          \"height\":\"\",
          \"phone\":\"189xxxxxxx\",
          \"create_time\":1547606496,
          \"update_time\":1547606577,
          \"device_infos\":[
            {
              \"device_id\":\"39175\",
              \"device_type\":1,
              \"remark\":\"\",
              \"device_name\":\" \\u6536\\u94F6\\u53F0\",
            }
          ]
        }
      ]
    }
  }"
}
```

```
        "ext_setting": "",
        "device_shift_type": 2,
        "sn_code": "",
        "bind_status": 1,
        "bind_qrcode_source_type": 0,
        "bind_qrcode_id": "",
        "bill_channel_id": "",
        "out_device_id": "sz01T80575"
    }
],
"staff_infos": [
    {
        "staff_id": "113080",
        "staff_name": " \u5c0f\u5b9d",
        "remark": "",
        "shop_manager": true,
        "receive_one_code_pay_notify": true,
        "mch_open_id": "ouNh7wEz3JY-k7pyvEGDXZHI9bfU",
        "cloud_pay_open_id": "oIEuZ07nip9kkTfjD9LjNCyR7Qd0",
        "trade_qr_code": "https://pay.qcloud.com/cpay/qrcode_person?
id=sz01HPuVzJ3mW3b24JpU&out_shop_id=sz01fHOHajay0TrGcQDk&rid=11",
        "refund_auth": false,
        "national_code": "86",
        "mobile_phone": "133xxxxxxx",
        "out_staff_id": "sz01T119988"
    }
],
"out_shop_id": "sz01fHOHattyw4IXNbve",
"out_shop_id_url": "https://pay.qcloud.com/cpay/qrcode_shop?id=sz01fHOHattyw4IXNbve&rid=11",
"fee_type": "CNY",
"goods_tag": "",
"receive_notify_after_duty": true,
"pay_page_mode": 1,
"shop_type": 1,
"qrcode_infos": [
    {
        "qrcode_id": "SZ01PPU9HKE52G9JIUbx",
        "qrcode_name": "02",
        "type": 1,
        "staff_id": ""
    },
    {
        "qrcode_id": "SZ01OTVWV2127M3MPPde",
        "qrcode_name": "01",
        "type": 1,
        "staff_id": ""
    }
],
"send_micro_pay_notify": true,
"shop_group_id": "",
"disable_membership_card": false,
"wxpay_shop_info_ext": {
    "ext_sub_mch_id": ""
},
"alipay_shop_info_ext": {
    "ali_authorization_url": "https://pay.qcloud.com/cpay/alipay_mch_authorize?
oshop=sz01fHOHattyw4IXNbve&rid=11"
}
},
],
```



```
    "total_count":1,
    "nonce_str":"juSTKFMz80nmRUypBeQBteU0eAHbbXyF"
  },
  "authen_info":{
    "a":{
      "authen_type":1,
      "authen_code":"5434DDE1D6CDAFBAC179B96E40D346455F518C7558F04C35FDF30AAB0278D420"
    }
  }
}
```

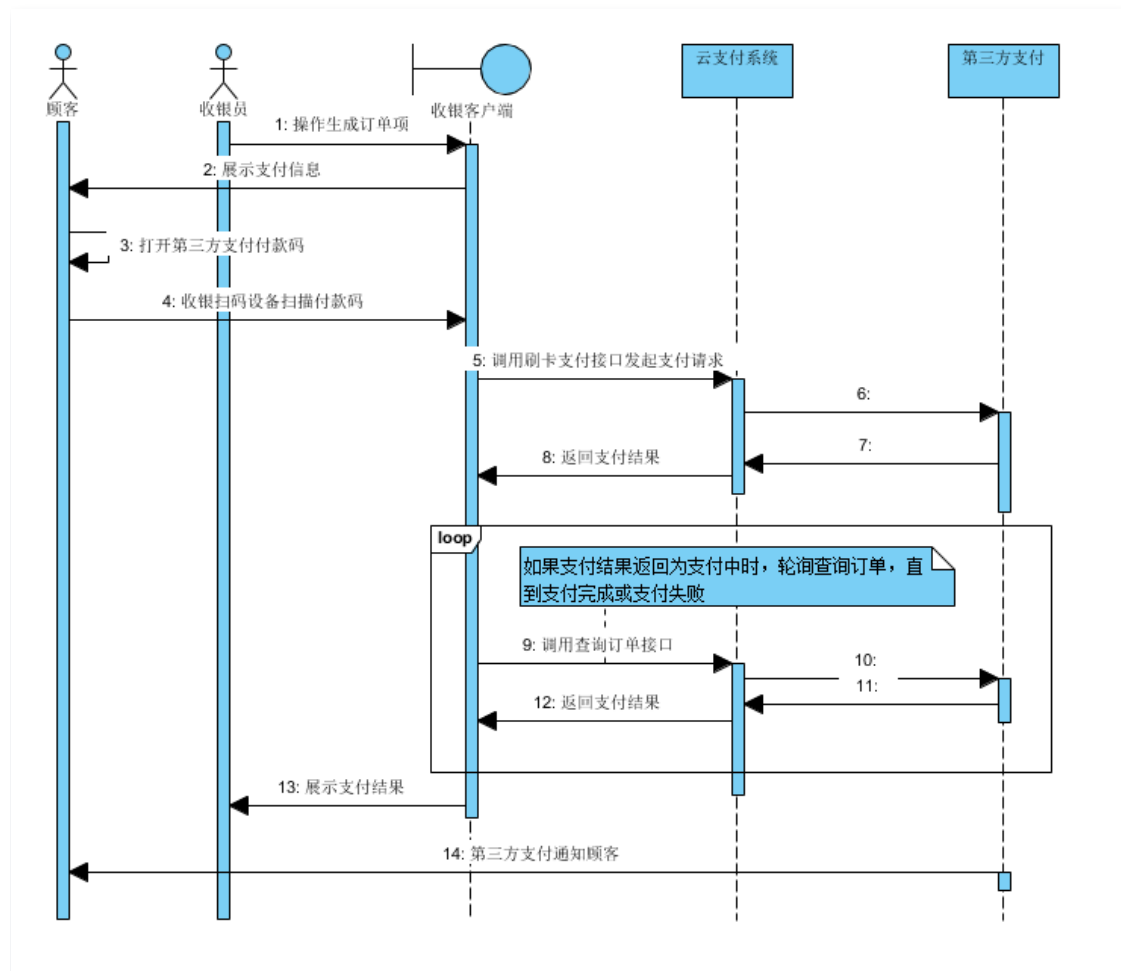
支付场景

刷卡支付场景

最近更新时间：2020-06-29 16:45:29

本节主要介绍刷卡支付相关接口使用场景。

整个刷卡支付场景序列图如下：



主要步骤：

- 第1 - 2步收银员根据订单确定收款金额。
- 第3步顾客打开第三方支付平台付款码。
- 第4步收银员扫描取得用户付款码。
- 第5步收银客户端提取付款码，生成订单号以及 **商户其它信息**，调用 **刷卡支付** 接口 `micro_pay` 请求云支付系统。
- 第6 - 8步返回支付结果。
 - 如果第三方支付平台支付状态为支付中，则执行第9 - 12步。
 - 如果未支付完成跳至第13步。
 - 如果支付失败则直接跳到第14步。
- 第9 - 12步轮询调用 **查询订单** 接口 `query_order`，直到第三方支付平台返回支付成功或支付失败。如果2分钟内没有轮询到支付结果，请收银员去云支付公众号平台确认支付结果。
- 第13步顾客收到第三方支付平台通知。
- 第14步收银客户端给收银员展示支付结果。

⚠ 注意

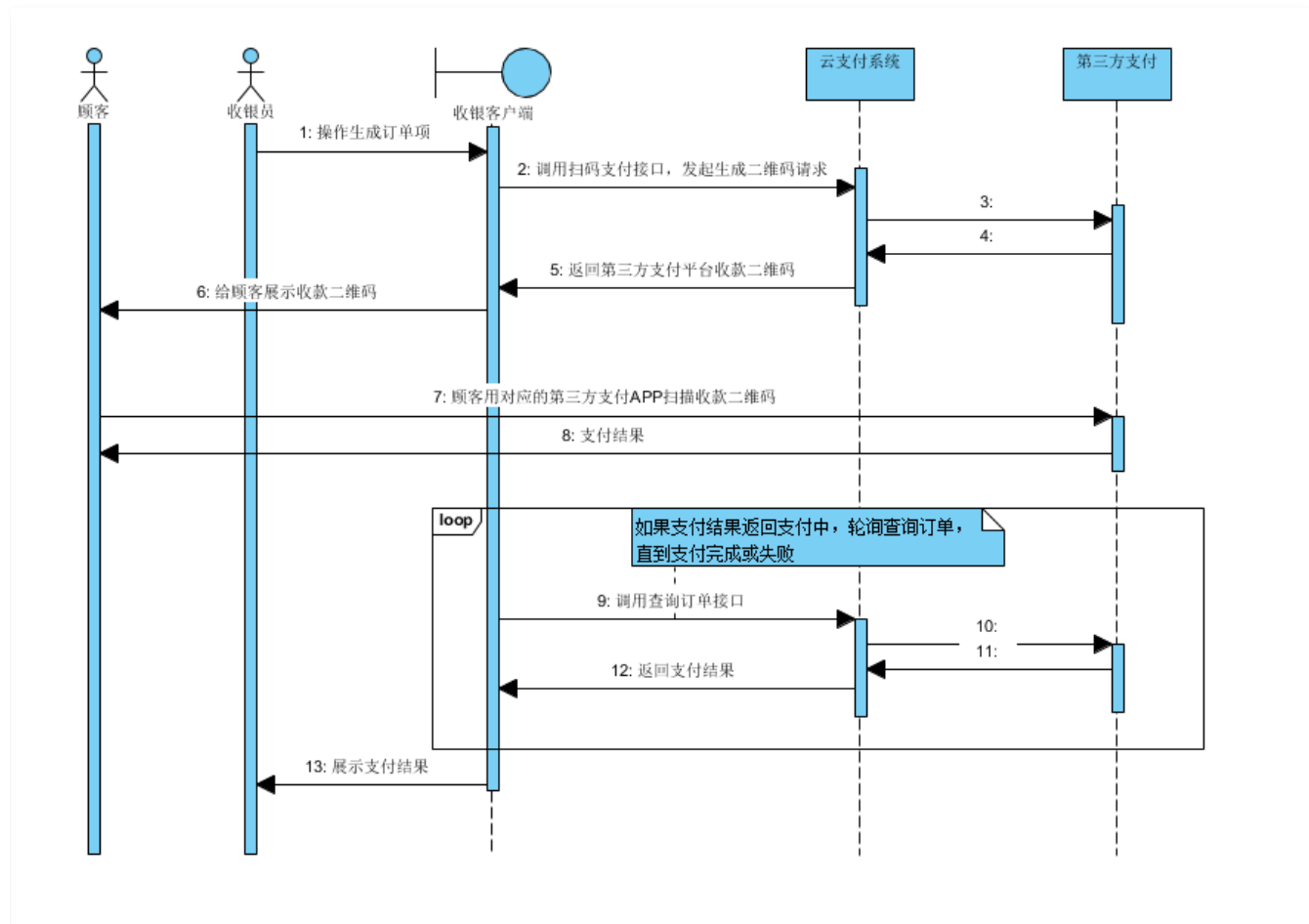
接口调用方要保证订单号在商户下唯一。

扫码支付场景

最近更新时间：2020-06-29 16:47:17

本节主要介绍扫码支付相关接口使用场景。

整个扫码支付场景序列图如下：



主要步骤：

- 第1步收银员根据订单确定收款金额，并与顾客确定使用的第三方支付平台。
- 第2步收银客户端根据收款金额、第三方支付平台、生成订单号以及 **商户其它信息**，调用 **扫码支付** 接口 `scan_code_pay` 请求收款二维码。
- 第3 - 5步收银客户端根据上一步的请求结果提取 `code_url`。
- 第6步在面向顾客的屏幕上将 `code_url` 转化成收款二维码。
- 第7 - 8步顾客打开第三方支付 App 扫描收款二维码并支付。
- 第9 - 12步轮询调用 **查询订单** 接口 `query_order`，直到第三方支付平台返回支付成功或支付失败。如果2分钟内没有轮询到支付结果，请收银员去云支付公众号平台确认支付结果。
- 第13步收银客户端给收银员展示支付结果。

⚠ 注意

- 接口调用方要保证订单号在商户下唯一。
- 第9 - 12步可以在生成二维码之后就开始轮询。

SDK 说明

SDK 简介

最近更新时间：2021-05-20 17:51:24

简介

云支付 SDK 对支付请求的各种参数进行打包和签名，封装了各种使用上的细节，并包含了自动重试、运行日志上传等功能，服务商使用更方便，且定位故障更快速、准确。

SDK 接口

SDK 提供了普通初始化、刷卡支付、扫码支付、查询订单、申请退款、查询退款、门店下载、订单批量下载、交易统计等接口。

第三方支付平台

SDK 支持微信支付和支付宝。

SDK 平台和语言

- Windows 环境下的 C++
- C
- .NET C# (.net 的版本要支持 tls1.2，.net 版本要 >= 4.6)
- Java

下载地址

- **Windows 环境 C++**：其中刷卡支付和扫码支付为异步方式，接口调用成功只代表支付提交成功，支付结果需要通过查询订单获取；取消订单、申请退款、退款查询、门店上传、门店下载接口为同步调用云支付接口。解压后包含以下两个目录：
 - CloudPayAPI_SDK_CPP_DLL 目录，包含编译好的 dll 库，可直接使用。
 - CloudPayAPI_SDK_CPP 目录，包含源码，开发者可自行编译。
- **C**：所有接口均为同步接口。
- **C#**：所有接口均为同步接口。
- **Java**：所有接口均为同步接口。

SDK 接入流程

最近更新时间：2021-09-06 14:57:04

微信云支付 SDK

SDK 支持 C/C++/C#/Java。SDK 说明请参见 [SDK](#)。

接入场景

本文为您介绍收银机+扫码枪场景的接入流程，在收银机上实现腾讯云支付功能，当用扫码枪扫码顾客微信支付或支付宝付款码，收银机调用云支付的 API 发起刷卡支付。



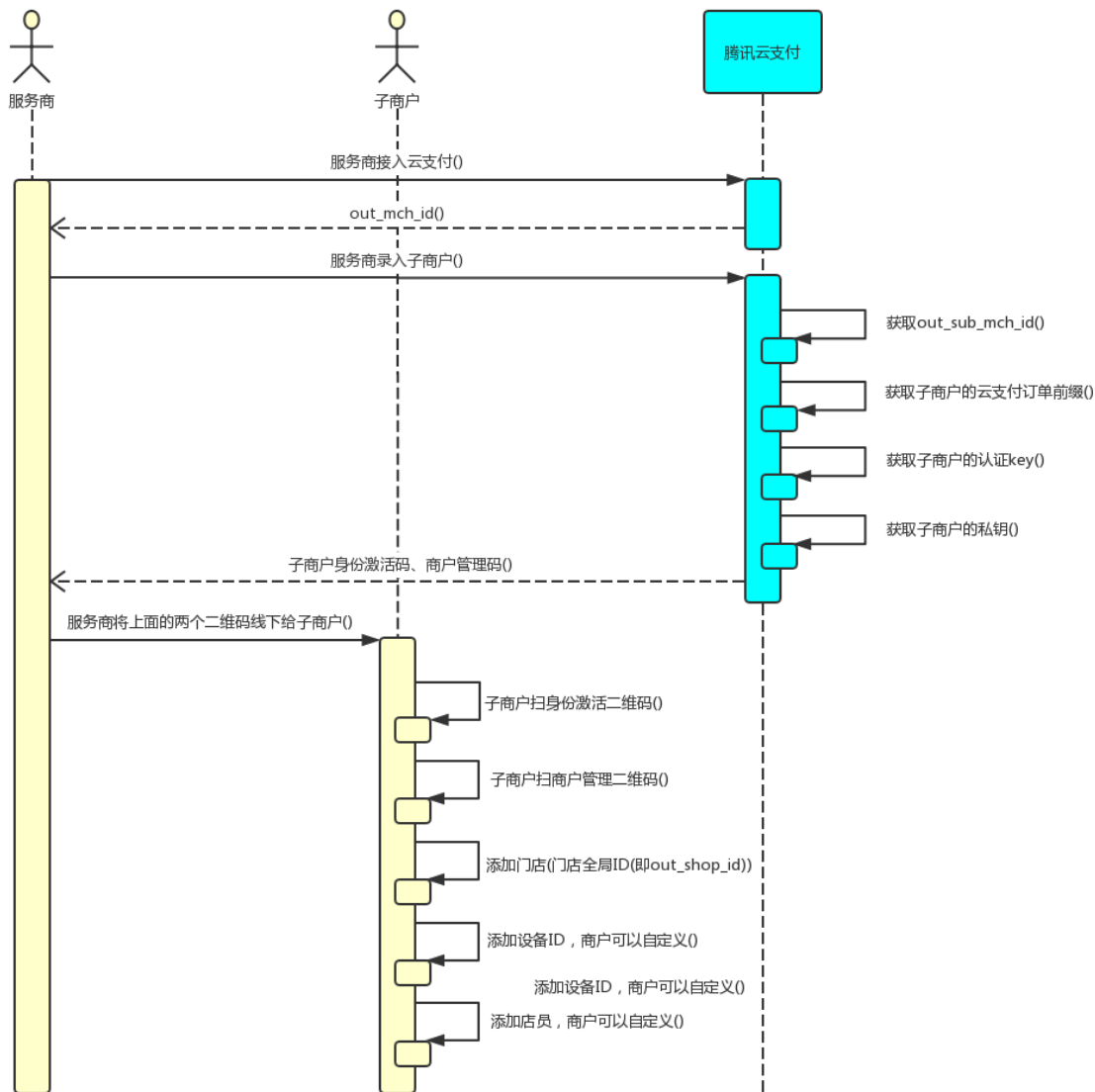
接入流程

服务商/子商户入驻流程

前提条件

登录 [腾讯云官网](#) PC 端录入服务商、子商户、门店，详细操作请参见 [配置服务商](#)、[配置子商户](#)、[配置门店](#)。

入驻流程



说明

流程图中的 out_mch_id、out_sub_mch_id、out_shop_id、decive_id、staff_id、子商户订单前缀、认证 key、私钥在调用刷卡支付时需用到。

获取云支付服务商账号

登录 [云支付控制台](#)，选择**服务商账户管理**，在服务商账号列表可查看云支付服务商账号。

获取云支付子商户账号

登录 [云支付控制台](#)，选择**子商户管理**，在子商户账号列表可查看云支付子商户账号。

获取订单号前缀

在子商户详情页有两个订单前缀，推荐使用纯数字的8位订单前缀，便于机具扫订单条码退款时很容易扫上。

获取门店全局 ID、设备 ID（设备编号）、店员 ID（店员编号）

×

门店信息

...

收款码

☰ >

门店ID

1

门店名

南山店

门店电话

13666668833

订单优惠标记

测试123

门店全局ID

sz01qyoPJmd3j1hWmul4

口碑门店

辰茂皇庭大酒店中餐水头店 >

微信买单门店

>

支付宝子商户信息

>

微信支付子商户号

>

币种类型

人民币(CNY) >

支付模式

默认模式 >

<

>

×

编辑设备

...

设备信息

设备编号

1

设备名称

记录

收银类型

刷卡支付 >

删除设备

保存

<

>

×

编辑店员信息

...

员工信息

收款码

☰ >

店员编号

1234

店员名称

test1

手机号

13800138000

测试

<

2/64

设置收银终端密码

>

是否为店长

☒

接受门店订单支付通知

☒

删除店员

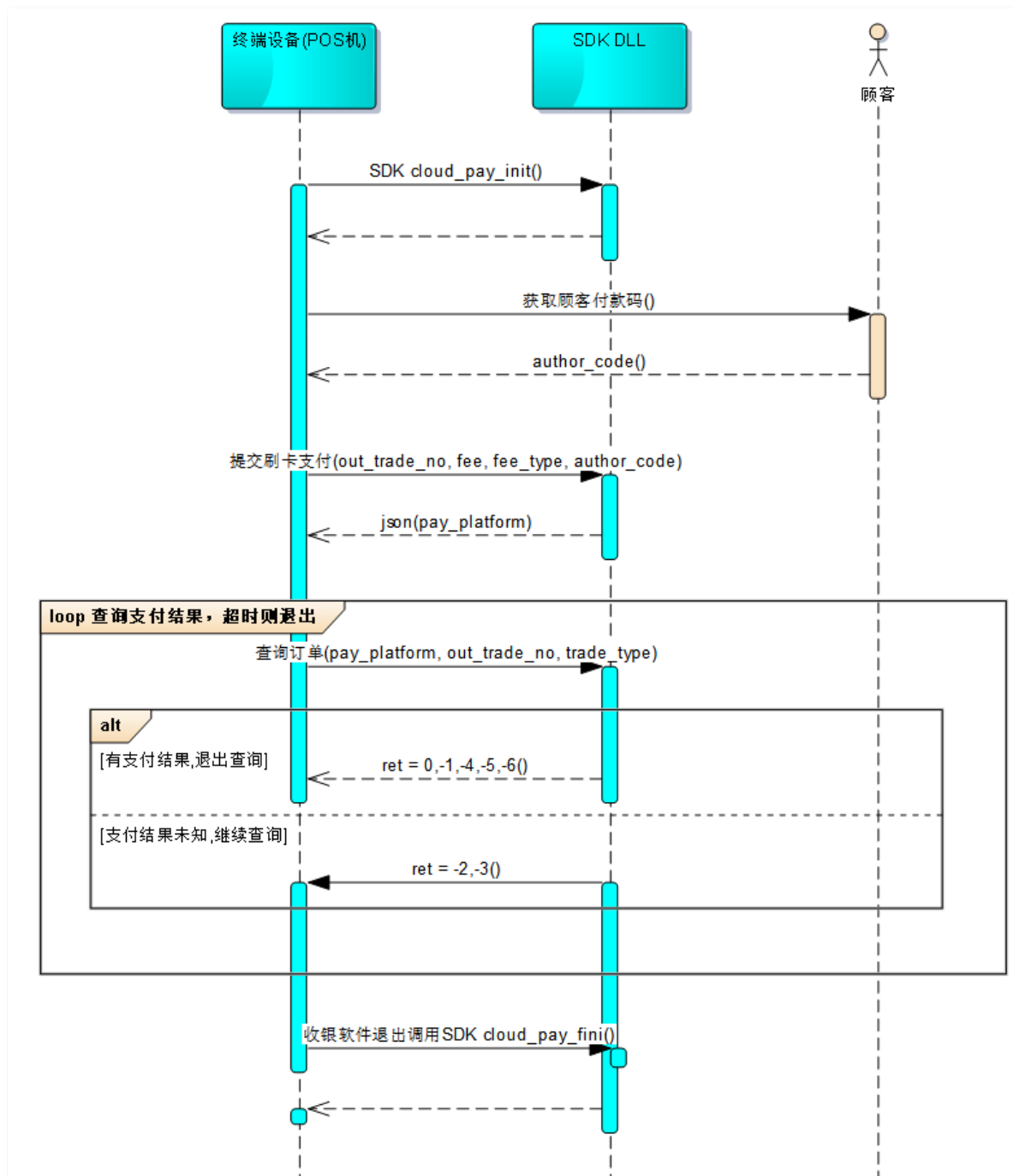
<

>

- 获取认证密钥和签名私钥
配置子商户时，保存到本地的“认证密码”和“签名私钥”。

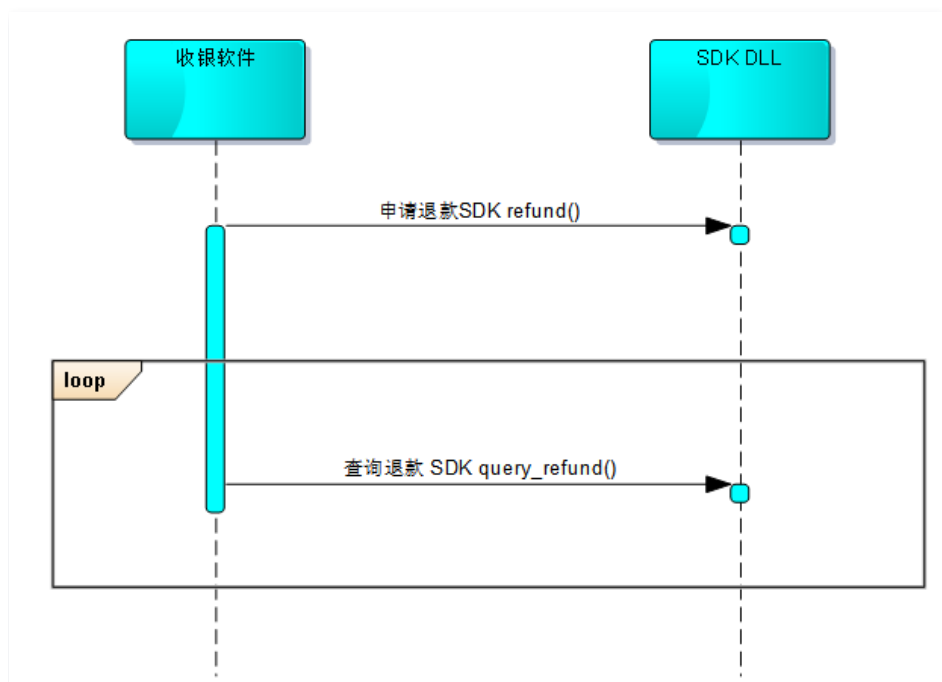
SDK API 接入流程

刷卡支付接入流程（顾客被扫）



1. 初始化 SDK 时, commonconf 配置的 micro_pay_query_order_count 和 micro_pay_query_order_interval 两个参数, 分别为刷卡支付后查询结果次数和间隔, 默认次数是25次、时间间隔是6s, 即一共会持续查询150s看是否有支付结果。
用户如果需要调整次数和间隔时间太, 可在初始化时设置这两个参数的值。
2. 提交刷卡支付时, out_trade_no 为云支付分配的子商户订单前缀+商户自己定义的部分。子商户订单前缀可在云支付控制台的子商户页面获取。
3. 刷卡支付是一个异步操作, 提交刷卡支付成功, 不代表支付成功, 只代表 SDK 已经受理这笔交易, 调用方需要调用查单接口来确认支付结果。
4. 刷卡支付提交后, 返回参数里会有一个 json 结构的支付平台, 需要调用方解析出来, 查单的时候需用这里解析出来的支付平台。
5. SDK 提供的接口, 请求参数和返回参数均是 json 结构。
6. 使用 SDK 的 get_errmsg() 接口获取具体的错误描述信息。

退款接入流程



- 申请退款成功，并不代表退款成功。
退款是一个异步过程，申请退款成功后，只是代表第三方支付平台受理这笔退款，是否到账需要看这笔钱退到哪里。
- 如果是退款至微信零钱账户或支付宝账户，会很快到账。
- 如果是退款至顾客的银行卡，到账时间与银行处理进度相关，可能要花几个小时。
因此申请退款成功后，需告知顾客，申请退款已经成功，请关注后续到账情况。顾客可以关注微信或支付宝的消息，查看已经发起退款或退款到账的消息。
- 如果想在设备上看这笔退款是否到账，可以调用退款查询接口，通过退款单的状态来查看是否退款成功。

日志上报

接口说明

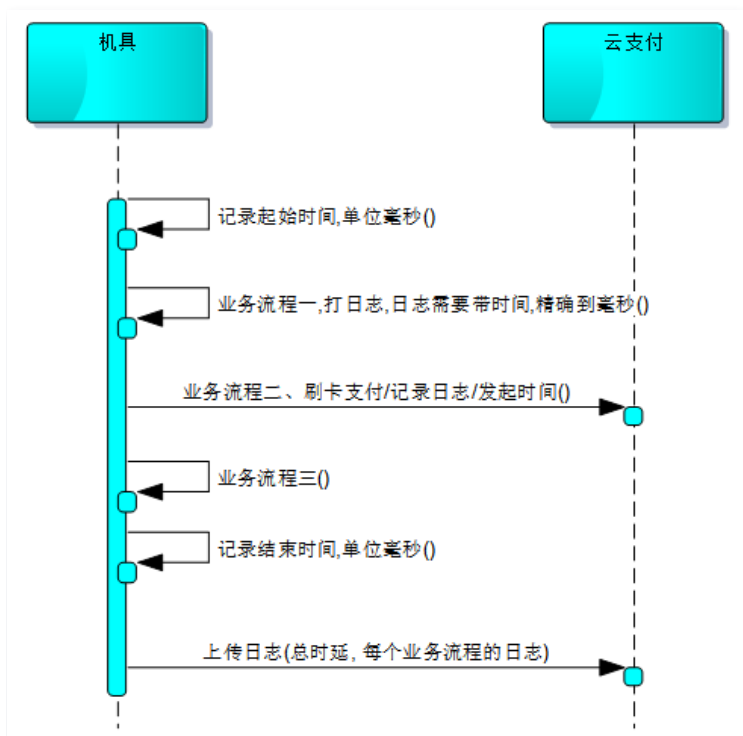
机具直接对接云支付的情况下，为方便商户定位问题，云支付提供日志上报接口，将日志数据上报到云支付的后端，可以帮助商户分析问题，例如机具业务流程各个环节消耗了多少时延，业务流程各个环节的数据情况等。

接口名：https://pay.qqcloud.com/cpay/upload_client_monitor_info

! 说明

- 云支付日志上报是接口维度的，例如刷卡支付、查询订单、退款。每个接口，调用方可以统计调用接口的时延传到云支付。
- 业务流程上的日志，可以上传多条，每天日志的记录时间点、精确到毫秒，需要调用方自己来完成。

流程说明



说明

总时延 = 记录结束时间 - 记录起始时间

设备维度订单统计

商户需要按设备维度统计订单相关的信息时：

- 统计某一个设备的订单信息，需要接入方知会商家在移动端管理系统，添加门店的设备 ID 和设备名称。
- 统计某一个款设备型号的订单，需要在支付时，OrderClient 结构里填写 sub_terminal_type 字段。
- 统计某一个平台的订单，例如 Windows 平台、Andriod 平台，需要在支付时，OrderClient 结构里填写 terminal_type 字段。

常见问题

请求格式

以下是刷卡支付的请求内容，为方便用户查看需注意点，以下一整个字符串换行显示。

```

1 {
2   "request_content": {
3     "nonce_str": "416492026bc84091bcaf7e74ea90ceba",
4     "order_client": {
5       "device_id": "1",
6       "machine_no": "P2L00000015232400422",
7       "sdk_version": "1.1",
8       "spbill_create_ip": "183.15.244.75",
9       "staff_id": "1003",
10      "terminal_type": 2,
11      "pay_content": {
12        "author_code": "134680423163089456",
13        "body": "生活用品套餐",
14        "fee_type": "CNY",
15        "out_trade_no": "sz01001mnx20171228151031",
16        "total_fee": 1,
17        "pay_mch_key": {
18          "out_mch_id": "sz013NzuonO6CMJd0rCB",
19          "out_shop_id": "sz01qyoPjmd3j1hWmul4",
20          "out_sub_mch_id": "sz01ELTR2810FpmdAp6J",
21          "pay_platform": 1
22        }
23      }
24    },
25    "authen_info": {
26      "a": {
27        "authen_code": "373FD1BD9D18A4C60C88BA9B62FBB63370CCD68CBEA47381809C5843254DB3D",
28        "authen_type": 1
29      }
30    }
31  }

```

格式注意一

整个数据格式是一个 json，其中 request_content 的内容也是一个 json，因此这里有两层，所以第二层有\出现。如下图，括号里的内容本身也是一个 json 结构，toString 后赋给 request_content。

```

1 {
2   "request_content": "{
3     \"nonce_str\": \"416492026bc84091bcaf7e74ea90ceba\",
4     \"order_client\": {\"device_id\": \"1\", \"machine_no\": \"P2L00000015232400422\",
5     \"pay_content\": {\"author_code\": \"134680423163089456\", \"body\": \"生活用品\",
6     \"pay_mch_key\": {\"out_mch_id\": \"sz013NzuonO6CMJd0rCB\", \"out_shop_id\": \"sz013qyoPJmd3j1hWmul4\"},
7   },
8   \"authen_info\": {\"a\": {\"authen_code\": \"373FD1BD9D18A4C60C88BA9B62FBB63370CCCD68CBEA\"}}
9 }

```

格式注意二

request_content 的内容是一个字符串，如下图，所以第2行和第7号，分别有一个双引号。

```

1 {
2   "request_content": "{
3     \"nonce_str\": \"416492026bc84091bcaf7e74ea90ceba\",
4     \"order_client\": {\"device_id\": \"1\", \"machine_no\": \"P2L00000015232400422\",
5     \"pay_content\": {\"author_code\": \"134680423163089456\", \"body\": \"生活用品\",
6     \"pay_mch_key\": {\"out_mch_id\": \"sz013NzuonO6CMJd0rCB\", \"out_shop_id\": \"sz013qyoPJmd3j1hWmul4\"},
7   },
8   \"authen_info\": {\"a\": {\"authen_code\": \"373FD1BD9D18A4C60C88BA9B62FBB63370CCCD68CBEA\"}}
9 }

```

计算认证码

参与计算认证码的数据内容如下图，双引号内的数据参与计算认证码，request_content 这个关键字不参与计算认证码。

```

1 {
2   "request_content": "{
3     \"nonce_str\": \"416492026bc84091bcaf7e74ea90ceba\",
4     \"order_client\": {\"device_id\": \"1\", \"machine_no\": \"P2L00000015232400422\",
5     \"pay_content\": {\"author_code\": \"134680423163089456\", \"body\": \"生活用品\",
6     \"pay_mch_key\": {\"out_mch_id\": \"sz013NzuonO6CMJd0rCB\", \"out_shop_id\": \"sz013qyoPJmd3j1hWmul4\"},
7   },
8   \"authen_info\": {\"a\": {\"authen_code\": \"373FD1BD9D18A4C60C88BA9B62FBB63370CCCD68CBEA\"}}
9 }

```

请求包完整示例

```

Json::Value pay_mch_key;    // 构造 pay_mch_key
pay_mch_key["pay_platform"] = 1;
pay_mch_key["out_mch_id"]   = "sz013NzuonO6CMJd0rCB";
pay_mch_key["out_sub_mch_id"] = "sz01ELTR281OFpmdAp6j";
pay_mch_key["out_shop_id"]  = "sz01qyoPJmd3j1hWmul4";

Json::Value pay_content;   // 构造 pay_content
pay_content["out_trade_no"] = "sz0100Imnx20171228151031";
pay_content["author_code"]  = "134680423163089456";
pay_content["total_fee"]    = 1;
pay_content["fee_type"]     = "CNY";
pay_content["attach"]       = "attach";

Json::Value order_client;  // 构造 order_client
order_client["machine_no"]  = "32-62-A8-14-B3-C0";
order_client["sdk_version"] = "1.0";
order_client["device_id"]   = 1;
order_client["spbill_create_ip"] = "10.15.244.75";
order_client["staff_id"]     = "1003";
order_client["terminal_type"] = 2; // 平台 Windows Linux Android pos

```

order_client["sub_terminal_type"] = 1111; //设备型号，如商家的 pos 的 AXX01 型号，这个字段接入方自定义，保证自己设备型号的唯一性，可用于统计某一款设备的订单信息。

```
Json::Value request_content;    // 构造 request_content
request_content["pay_mch_key"] = pay_mch_key;
request_content["pay_content"] = pay_content;
request_content["order_client"] = order_client;
request_content["nonce_str"] = "416492026bc84091bcaf7e74ea90ceba";

Json::FastWriter w;
std::string request_content_str = w.write(request_content);

Json::Value authen;
authen["authen_code"] = hmac_sha256(authen_key, request_content_str); //计算认证码
authen["authen_type"] = 1; //hmac_sha256 为1

Json::Value authen_info;
authen_info["a"] = authen; //认证码，签名是 s

Json::Value request;    //构造最终发给服务器的请求
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;

std::string request_str = w.write(request);

return request_str;
```

订单号

- 不同设备的唯一性

订单号由云支付分配的订单唯一前缀+商户自定义两部分组成。其中商户自定义部分需要保证不同设备的唯一性，例如一个商户的订单号自定义部分为年+月+日+时+分+秒，会在多个机器支付时出现订单号重复，导致支付失败。

- 订单号不能回滚

例如商户自定义部分订单号为：机号+年月日+序列号，序列号由1、2、3、.. 往上增。如果某个时间点机器故障或软件故障，序列号不能又从1、2、3往上增，否则会出现订单号重复导致支付失败。

机具接入说明

机具对接流程

最近更新时间：2021-09-06 14:53:24

对接流程

HTTPS API 说明请参见 [HTTPS API](#)。

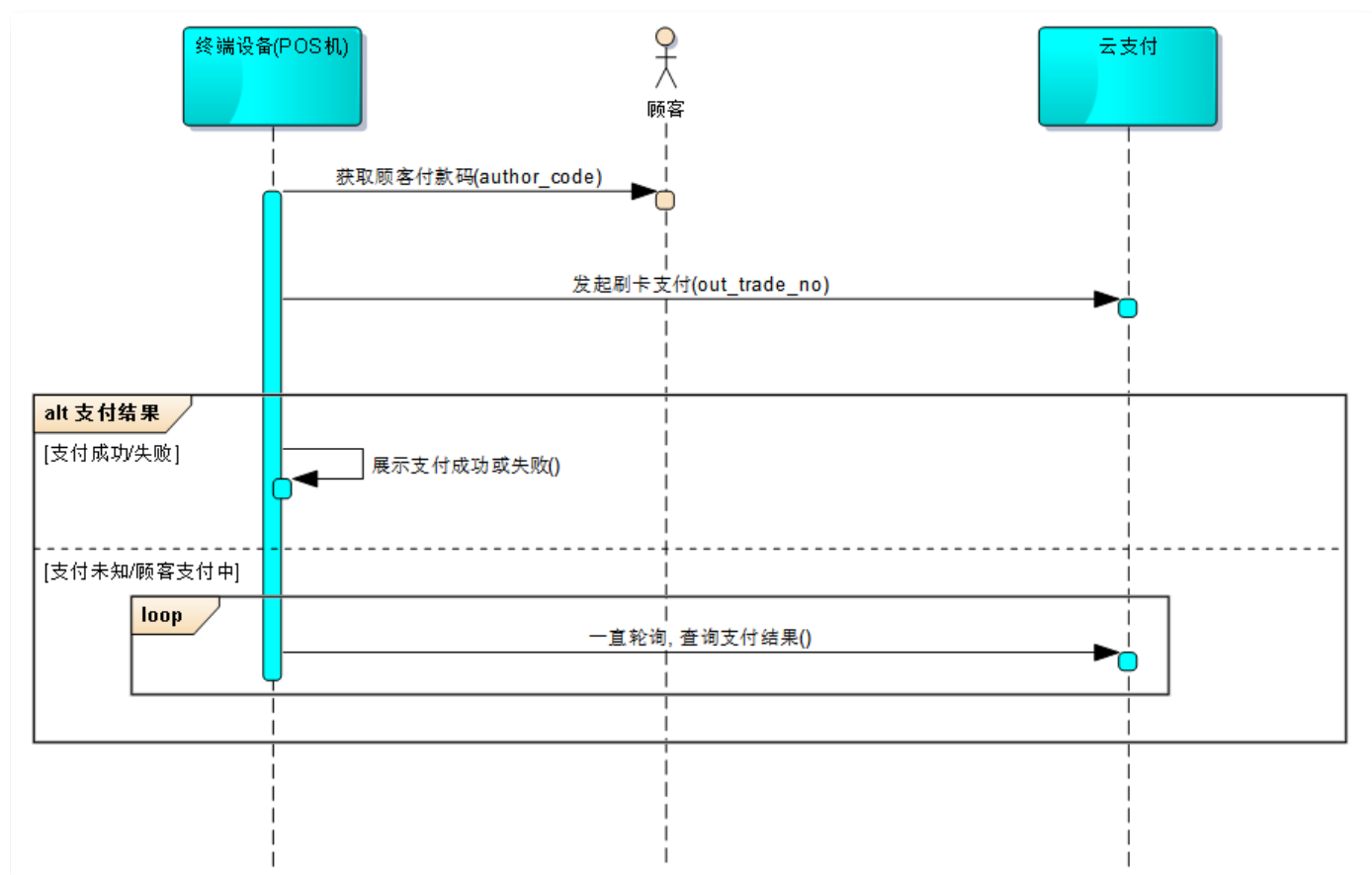
⚠ 注意

其中请求方的 HTTPS 握手协议需是 TLS1.1 及以上的协议，否则连接会失败。

服务商/子商户入驻总流程请参见 [服务商和子商户配置（新版）](#)。

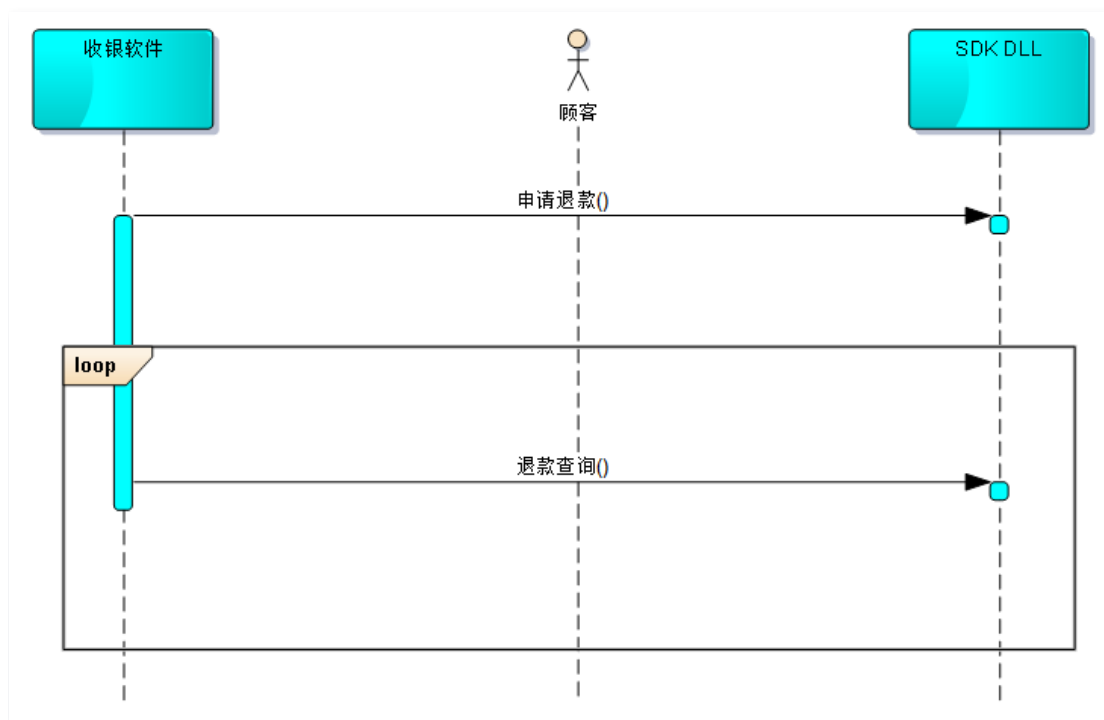
HTTPS API 交易流程

刷卡支付接入流程（顾客被扫）



- 刷卡支付接口：https://pay.qcloud.com/cpay/micro_pay。
- 查询订单接口：https://pay.qcloud.com/cpay/query_order。
- 支付结果未知包含：网络请求超时和业务上的结果未知，详情请参见 [HTTPS API](#) 对 status 的说明。
- 接口返回 status 为0，只表示业务请求成功，订单的支付结果需要通过订单的状态来判断。
- 如果2分钟内查询不到结果，请到手机端管理系统查看交易明细确认支付结果。

退款接入流程



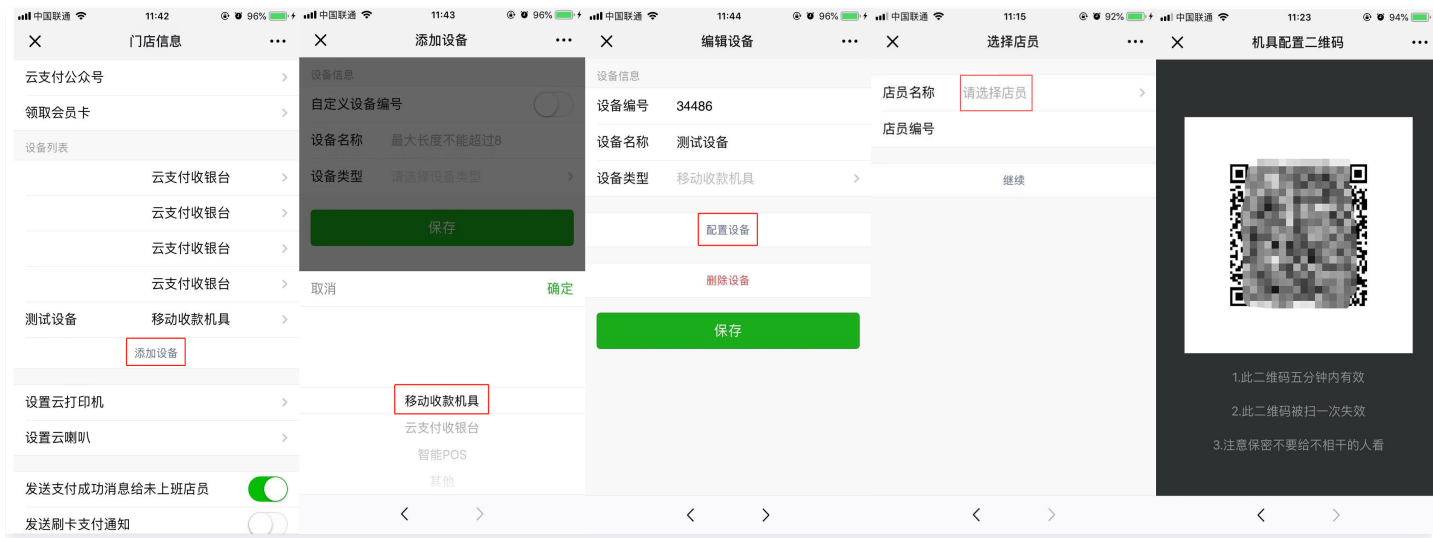
- 申请退款成功，并不代表退款成功。
退款是一个异步过程，申请退款成功后，只是代表第三方支付平台受理这笔退款，是否到账需要看这笔钱退到哪里。
- 如果是退款至微信零钱账户或支付宝账户，会很快到账。
- 如果是退款至顾客的银行卡，到账时间与银行处理进度相关，可能要花几个小时。
因此申请退款成功后，需告知顾客，申请退款已经成功，请关注后续到账情况。顾客可以关注微信或支付宝的消息，便能收到已经发起退款或退款到账的消息。
- 如果想在设备上看这笔退款是否到账，可以调用退款查询接口，通过退款单的状态来查看是否退款成功。适配者可以自己选择是否适配退款查询接口。

机具配置

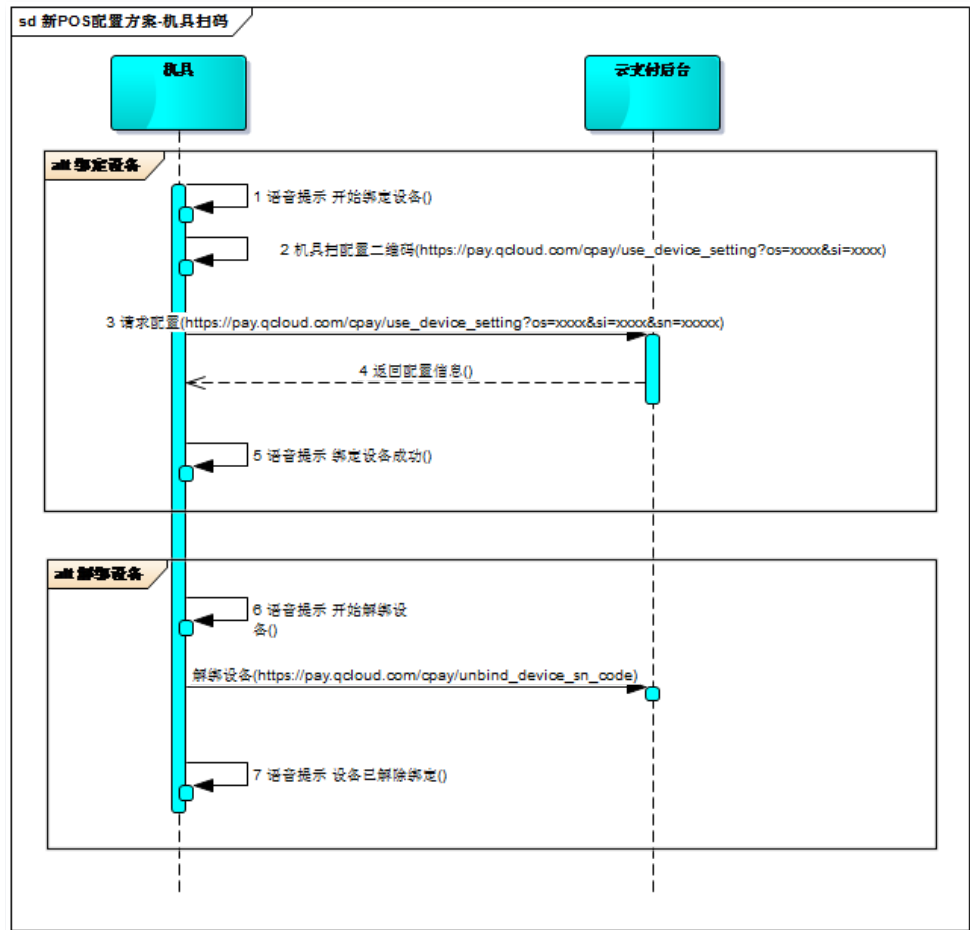
获取配置二维码

管理员登录手机商户管理系统，找到需要配置的设备，进行下图设置：

- 添加设备时，需选择**移动收款机具>配置设备**，完成后需选择店员，如果没有店员，需先创建店员再生成配置二维码。
- 生成的二维码链接如下：https://pay.qcloud.com/cpay/use_device_setting?osi=xxxx&si=xxxx，二维码的有效期是5分钟。
- 只有商户管理员能够获取配置二维码。



机具绑定/解绑流程



- 扫码配置二维码后，请求后台的 url 需要增加 sn 号，获取配置的请求为 HTTPS GET 请求。
- 配置二维码，有效期是5分钟。
- 机具绑定和解绑需要有语音提示。
- 机具需要支持重复配置，在配置异常时，就可以重新开始配置。

绑定配置接口说明

绑定配置 url

执行命令：GET https://pay.qcloud.com/cpay/use_device_setting?osi=xxxx&si=xxxx&sn=xxxxxx 。



请求是 get 请求，不是 post 请求。

请求参数

参数	含义	类型	长度	是否必填	备注
osi	全局门店 ID	String	20字节	必填	配置二维码中的参数 osi
si	云支付生成的配置 ID	String	20字节	必填	配置二维码中的参数 si
sn	sn 号	String	64字节	必填	sn 号为：厂家名称简称_型号_厂家 sn 号，其中厂家名称简称和型号都要求英文大写字母，保证不同厂家之间 sn 号唯一，以联迪为例 qm500 就是 LANDI_QM500_1923840238029389

响应参数

响应为一个 json 字符串，包含以下字段：

参数	类型	子参数	子参数类型	含义	长度	是否一定返回	备注
status	Number	-	-	错误码	-	一定	参见 HTTPS API 说明
description	String	-	-	错误描述	不超过 128 字节	一定	-
log_id	Number	-	-	日志 ID，用于定位问题	-	一定	调用方无需关注
internal_status	Number	-	-	详细错误码	-	一定	调用方无需关注
use_device_setting	json 结构	-	-	-	-	一定	-
-	-	out_mch_id	String	云支付服务商号	20 字节	一定	-
-	-	out_sub_mch_id	String	云支付子商户号	20 字节	一定	-
-	-	cloud_cashier_id	String	云支付商户订单前缀	8 字节	一定	8 位纯数字
-	-	authen_type	Number	认证类型	-	一定	参见 HTTPS API 说明
-	-	authen_key	String	认证密钥	32 字节	一定	-
-	-	out_shop_id	String	全局门店 ID	20 字节	一定	-
-	-	shop_name	String	门店名称	不超过 128 字节	一定	-

-	-	device_id	String	设备 ID	不超过 64 字节	一定	-
-	-	device_name	String	设备名称	不超过 128 字节	一定	-
-	-	staff_id	String	员工 ID	不超过 64 字节	一定	-
-	-	staff_name	String	员工名称	不超过 128 字节	一定	-

 说明

如果 internal_status 等于 3214，表示云支付的 device_id 已经绑定了其他 sn 号。

解绑配置接口说明

解绑配置 url

接口

https://pay.qcloud.com/cpay/unbind_device_sn_code

方法

POST 方法，content_type 为 application/json。

请求参数

参数	含义	类型	长度	是否必填	备注
request_content	请求内容	RequestContent	-	必填	详见本节 RequestContent；打包方式参见 HTTPS API 说明
authen_info	认证信息	AuthenInfo	-	必填	参见 HTTPS API 说明

RequestContent 结构

参数	含义	类型	长度	是否必填	备注
out_sub_mch_id	云支付分配给子商户的账号	String	20 字节	必填	-
out_shop_id	云支付唯一标识门店的账号	String	20 字节	必填	手机端管理系统显示的云支付全局门店 ID
device_id	云支付门店内的设备 ID	String	20 字节	必填	手机端管理系统门店内的设备 ID
sn_code	sn 号	String	64 字节	必填	调用绑定接口时使用的 sn 号

响应参数

响应为一个 json 字符串，包含以下字段：

参数	含义	类型	长度	是否必填	备注
response_content	响应内容	ResponseContent	-	必填	详见本节 ResponseContent；解析方式参见 HTTPS API 说明
authen_info	认证信息	AuthenInfo	-	必填	参见 HTTPS API 说明

ResponseContent 结构

参数	含义	类型	长度	是否必填	备注
status	错误码	Number	-	必填	参见 HTTPS API 说明
description	错误描述	String	256	必填	-
log_id	日志 ID	Number	-	必填	-
internal_status	具体错误码	Number	256	必填	厂商一般不需要关注具体错误码

❗ 说明

如果 internal_status 等于3215，表示云支付的 device_id 已经与原绑定设备解绑。

交易接口说明

- 在使用支付、退款、查询订单和查询退款单四个接口时，需要在请求的 OrderClient 结构里增加一个 sn_code 字段（类型 string），表示绑定设备的 sn 号，否则交易流程会走不通。
- 退款使用认证密钥计算认证码即可。

验收标准

必需功能

1. 支持配置

- 统一在云支付手机端管理系统配置。
- 支持绑定和解绑流程。
- 支持重新绑定；例如第一种情况：机具已绑定商户，在商户的认证密钥更新后需要重新绑定机具；第二种情况：手机端管理系统设备被删除后，机具需要重新绑定其他设备。

2. 支持连接方式

支持直连云支付。

3. 支持刷卡支付

- 需要输入密码，显示“等待用户输入密码中”。
- 支付结果未知，显示“通过订单查询确认支付结果或手机端商户管理系统确认支付结果”。

4. 支持订单退款

收银员扫订单条码，再查询订单得到订单总金额，会显示一个订单总金额作为默认的退款金额，如果需要部分退款，收银员修改金额即可。

5. 支持订单查询

云支付商户有2种订单号前缀，一种是带字符的10位，例如：sz0100lmnx；一种是纯数字的8位，例如：01000052，建议使用纯数字的8位前缀，便于机具扫描订单条码时容易扫上。

6. 支持店员可设置

7. 支持机具可升级

- 优先在线升级。
- 其他升级方式。

8. 支持多域名

机具需要支持多个域名切换，使用 /cpay/ping 接口可以探测域名是否能用。

云支付一共有如下4个域名：

pay.qcloud.com sz.pay.qcloud.com sh.pay.qcloud.com tj.pay.qcloud.com

第一种探测方式：例如列出4个域名，选择第一个，如果第一个不行，自动探测第二个，如果第二个不行，自动探测第三个，知道选择结果是否能用，给出相应的提示语。

第二种探测方式：支付的时候自动切换。

9. 支持扫码配置 WIFI

10. 支持 OrderClient 增加 sub terminal type 字段

云支付分配。

11. 支持打印小票

12. 支持语音播报

13. 机具操作手册

厂商提供服务商使用机具的操作手册。

可选功能

机具未提供以下3个功能时，收银员可在手机端商户管理系统查看配置。

1. 交易流水

2. 汇总统计

3. 语音区分支付平台

测试用例和注意事项

刷卡支付

以下采用微信支付0.02，支付宝支付0.02测试，建议用户使用其他金额再次进行测试。

- 测试用户输入密码的情形（正常情况下，账号当日支付超过10笔开始要求输入密码）。

需注意的边界情况：

- 不能语音播报超时但用户支付成功。
- 支付金额0（包括输入0.001等四舍五入后为0的）应在机具层面拦截，并提示用户。

退款和查单

以下流程对微信支付、支付宝分别进行测试：

- 支付0.02，应支付成功。
- 扫订单号条码，应显示支付成功。
- 退款0.03，应在机具层拦截，并提示超过可退金额。
- 继续退款0.01，应退款成功。
- 扫订单号条码，应显示订单状态为已部分退款。
- 继续退款0.02，应拦截并提示超过可退金额。
- 继续退款0.01，应退款成功。
- 扫订单号，订单状态应显示已全额退款。

⚠ 注意

- `current_trade_state` 字段只有全额退款时才会变为4，否则一直为2（成功），所以不能通过 `query_order` 接口判断支付宝订单是否发生过部分退款。
- 使用 `client_order_detail` 查订单得到订单的 `refunded_fee` 值，才能判断是否已经部分退款，但 `client_order_detail` 接口目前返回报文大小可能超过10k，请注意机具的处理能力，防止宕机。

语音播报

支付和退款都能正确播报平台和金额，应注意播报用户需要的结果信息，而不仅是播报“成功”或“查询成功”。

其他

必须正确填写 `sub_terminal_type` 字段，每个品牌机具取值区间需在云支付分配，请勿随意填写或不填。

收款机具 API 接入流程

最近更新时间：2023-08-17 10:35:12

1. 修订记录

版本号	变更内容	变更日期
1.0	创建文档	2019-10-06

2. 概述

机具通过小程序绑定指定商户与门店后，便可收款。机具的绑定和收款功能的后台能力由云支付提供。本文档主要说明了机具厂商如何通过云支付提供的 API，完成云支付后台的接入。

3. 机具要求

显示器能清晰展示 URL 为200字节大小的二维码。

4. 关键名词说明

4.1 sn 前缀

为了确保每个厂商机具的 sn 在云支付系统中不冲突，云支付会给每个厂商分配不同的 sn 前缀。
假设厂商按自有规则生成的原始 sn 为 91RA00001，云支付给厂商分配的 sn 前缀为 BF_M100_BF，其中 BF 为厂商简码，M100为产品型号简码，则需要存在两套新的 sn。

- 第一套为 BF91RA00001，即在原始 sn 前增加 BF。
- 第二套（完整 sn）为 BF_M100_BF91RA00001，即在原始 sn 前增加 BF_M100_BF。

调用云支付后台所有接口都使用第二套 sn。其它情况，如机具外壳上贴的 sn 都使用第一套 sn。

4.2 初始密钥

厂商需要给每台机具生成一个初始密钥，密钥是由数字（0-9）和大写字母（A-Z）组成的32字节随机字符串，而且必须是一机一密，不能相同。
调用云支付后台绑定相关接口需要使用这个密钥，详见 [机具绑定相关流程说明](#)。
此密钥要求高度保密，建议写入机具的安全区，至少要加密后再烧录到磁盘。由于初始密钥由厂商生成，厂商需要尽力保证密钥不外泄。

4.3 交易密钥

机具绑定成功后，会从云支付后台接口中获取到一个交易密钥。调用云支付后台支付相关接口需要使用这个密钥，详见 [机具收款相关流程说明](#)。
此密钥要求高度保密，获取到交易密钥后，必须先加密再写入磁盘，以防攻击者获取到这个交易密钥。

5. 接入准备工作

5.1 申请初始资源

厂商在正式进入开发阶段时，需要向云支付以邮箱（邮箱地址请咨询负责与厂商对接的人员）的方式申请获取以下内容。

名称	说明
sn 前缀	参见 4.1 sn 前缀 。
app_cpay_version	由云支付统一规划的机具程序版本号，绑定和数据上报时会用到。
sub_terminal_type	由云支付统一规划的设备类型码，支付的时候会用到。
加密用的公钥文件 public.key	由云支付统一发放的公钥文件，用于 sn 和初始密钥录入， 6.2 录入操作说明 会用到。
接收加密文件的邮箱地址	云支付接收 sn 和初始密钥信息的邮箱地址。

5.2 申请测试账号

默认情况下，厂商直接参考接口说明，在正式环境上调试接口即可。如遇后台接口还未正式发布等特殊情况，需要在测试环境调试接口，则联系负责与厂商对接的人员。

1. **sn 和初始密钥。** sn 和初始密钥是绑定等流程中必须的内容。如果在正式环境上调试，则需按第6章所述方法录入 sn 和初始密钥。
2. **测试商户。** 如果是正式环境测试，则调试人员需要有微信商业版的商户，通过“收款设备”小程序来完成绑定。

6. 机具出厂准备工作

6.1 背景说明

每台机具的 sn 和初始密钥录入到云支付后台后才能绑定成功，然后才能用于收款，初始密钥说明详见 [4.2 初始密钥](#)。

每批机具在出厂前，厂商需要把所有 sn 和初始密钥写入到一个文件中，并使用指定方法加密。通过厂商官方邮箱把加密文件发送给云支付指定邮箱，云支付收到邮件后会执行录入操作。录入完成后，云支付会回复邮件提示录入完成，厂商收到录入成功的邮件回复后，这批机具才能正式出厂。

厂商如果想调试绑定相关接口，需要先通这个方式把调试用到的 sn 和初始密钥录入云支付后台。

6.2 录入操作说明

1. 确认 openssl 版本

要求版本为 OpenSSL 1.1.0g 2 Nov 2017 及以上，可以通过 `openssl version` 命令查看版本号。

2. 生成 sn 和初始密钥文件

文件只包含两列，第一列为 sn（带完整前缀），第二列为初始密钥，中间以空格分隔，每行结尾不要有多余的空格或其它无关字符，中间不要有空行，最后一行不能为空行。生成的文件名为 `sn_init_key.txt`，文件内容示例如下：

```
BF_M100_BF91RA00001 696FF05FA7CE76664211A47CD788CC30
BF_M100_BF91RA00002 654502BB0F57EC8737B4CF12117575B8
```

3. 生成密码文件

随机生成一个32字节的密码字符串，并把这个密码字符串写入到一个名为 `encrypt_key.txt` 的文件。此文件只有一行，不能有多余的空格或空行。每次都要随机生成，不能使用相同的密码字符串。

4. 加密 sn 和初始密钥文件

使用第3步生成的 `encrypt_key.txt` 文件来加密第2步生成的 `sn_init_key.txt` 文件。执行命令：

```
openssl aes-128-cbc -salt -in sn_init_key.txt -out sn_init_key.txt.encrypted
```

这里会要求输入两次密码，密码内容即为第3步生成文件中的内容，直接从文件中复制过来就行，两次密码输入必须相同。执行成功后会生成一个 sn 和初始化密钥的加密文件 `sn_init_key.txt.encrypted`，即 `sn_init_key.txt` 加密后的文件。

5. 加密密码文件

使用 [5.1 申请初始资源](#) 所述云支付提供的公钥文件 `public.key` 加密第三步生成的密码文件 `encrypt_key.txt`。执行命令：

```
openssl rsautl -encrypt -inkey public.key -pubin -in encrypt_key.txt -out encrypt_key.txt.encrypted
```

执行成功后会生成一个密码文件的加密文件 `encrypt_key.txt.encrypted`。

6. 邮件发送加密文件

将加密文件发送到 [5.1 申请初始资源](#) 所述的接收加密文件的邮箱地址，需要发送的文件有 `sn_init_key.txt.encrypted` 和 `encrypt_key.txt.encrypted`。如果要文件打包进行发送，则必须使用 zip 来压缩。邮件发送时，抄送人需添加厂商相关管理人员。

7. 其它说明

以上所有文件都以 UTF-8 无 BOM 编码，发送前请先确认不能有可见/不可见的特殊字符。邮件发送完成后，销毁 `encrypt_key.txt` 和 `sn_init_key.txt`（无用销毁）文件，同时需保管好 `public.key` 文件，这些文件禁止外泄。

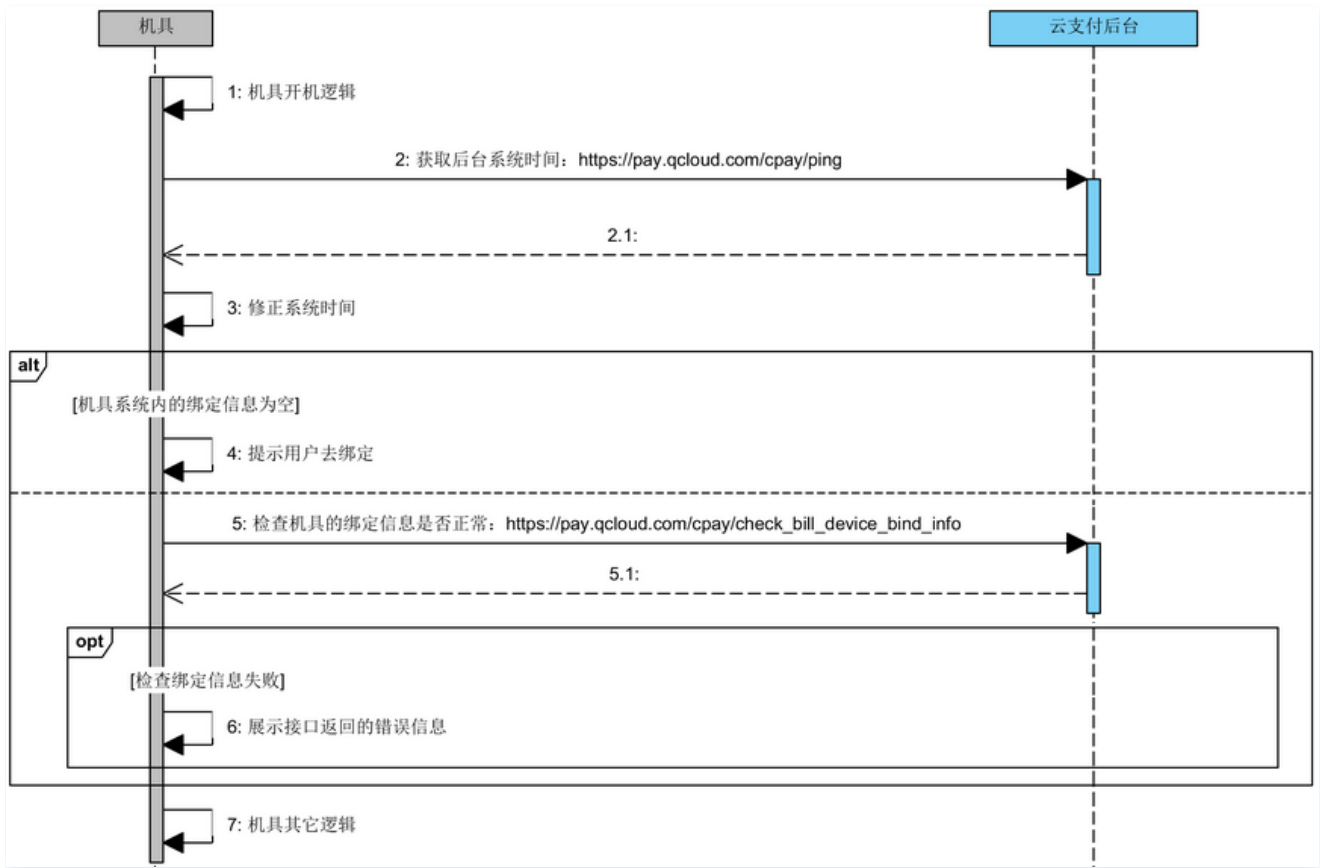
7. 机具绑定相关流程说明

7.1 背景说明

商户拿到机具后，需要先操作绑定才能进行收款。机具绑定主要是为了获取支付接口必须的部分参数内容，并在云支付后台建立逻辑绑定关系，从而成功发起收款。

7.2 机具开机或重启

以下为机具开机或重启成功后需要增加的逻辑处理过程：



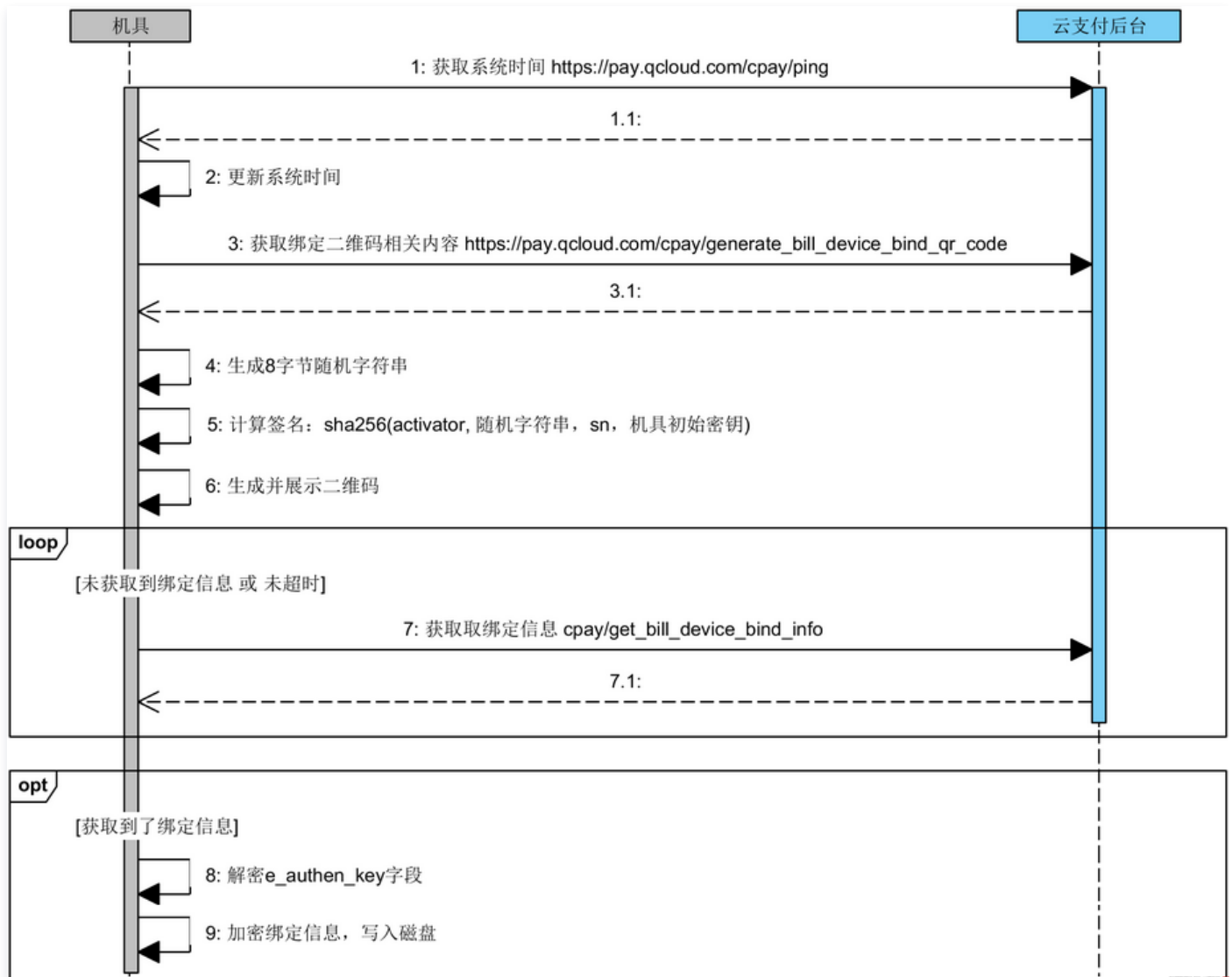
开机或重启完成后，调用云支付后台 ping 接口获取系统时间，并用获取到的时间修正系统时间。

- 如果机具系统内的绑定信息为空（没有绑定或恢复了出厂设置），则提示用户去绑定。
- 如果机具系统内的绑定信息不为空，则通过云支付后 check_bill_device_bind_info 接口来校验绑定信息是否可用，如果接口返回失败，则展示接口返回的错误信息。check_bill_device_bind_info 接口主要完成以下内容，这三件事如果检查通过，则可确保机具能正常收款。
 - 检查机具 sn 对应设备在云支付后台是否是“已绑定”状态；
 - 检查机具的在云支付后台的逻辑绑定关系是否正确；
 - 检查机具支付密钥是否和云支付后台保存的支付密钥一致。
- 显示屏展示对应提示后，用户可通过按键或其它方式关闭提示信息进行其它操作。

流程中涉及到的接口，详见 [接口说明](#)。

7.3 绑定

1. 时序图



机具增加“生成绑定二维码”的功能项，用户进入这个功能项后，执行时序图中的逻辑。

2. 获取绑定二维码信息

机具首先从云支付后台同步一次系统时间。然后通过 `generate_bill_device_bind_qr_code` 获取绑定二维码相关信息。

`generate_bill_device_bind_qr_code` 接口成功应答后，可以取到5个绑定逻辑相关字段：`activator`、`token`、`ttl`、`mini_program_url`、`interval`，接口详细说明详见 [接口说明](#)。

3. 生成二维码

二维码内容组成：`mini_program_url?ac=activator&rs=nonce_str&sign=sha256sum&sn=sn`

- `mini_program_url` 和 `activator` 来源于 `generate_bill_device_bind_qr_code` 接口应答。
- `nonce_str` 由机具生成的8字节随机字符串。
- `sign` 由机具使用初始密钥和多个字段拼接字符串（拼接格式：`ac=activator&rs=nonce_str&sn=sn`）计算的 sha256 值，即：
`hmac_sha256(ac=activator&rs=nonce_str&sn=sn, 机具初始密钥)`。

举例说明，假设：

- `mini_program_url` 值为：`https://payapp.weixin.qq.com/smartqr/entry/home`。
- `activator` 值为：`p9fa01zex4mi`。
- 机具完整 `sn` 值为：`LANDI_QM600_LA190000000`。
- 机具初始密钥值为：`C852274D372982D5530B692FD77124D3`。

则二维码生成流程为：

- 机具生成的8字节随机字符串（`nonce_str`）：`c44f32e0`。

4. 机具拼接字符串 `ac=p9fa01zex4mi&rs=c44f32e0&sn=LANDI_QM600_LA190000000`。

5. 机具计算 sha256: hmac-sha256(“ac=p9fa01zex4mi&rs=c44f32e0&sn=LANDI_QM600_LA190000000”, “C852274D372982D5530B692FD77124D3”), 获得 HMAC-SHA256 结果: E0CA8FDA56464BDF702FCA1AE9692EB99483E065879247763B3A46B576EBF5AC。

6. 拼接二维码 URL:

https://payapp.weixin.qq.com/smartqr/entry/home?

ac=p9fa01zex4mi&rs=c44f32e0&sn=LANDI_QM600_LA190000000&sign=E0CA8FDA56464BDF702FCA1AE9692EB99483E065879247763B3A46B576EBF5AC

机具显示器把拼接二维码 URL 展示成二维码。

7. 轮询绑定结果

机具展示出绑定二维码后, 立即开始定时轮询云支付后台绑定结果。

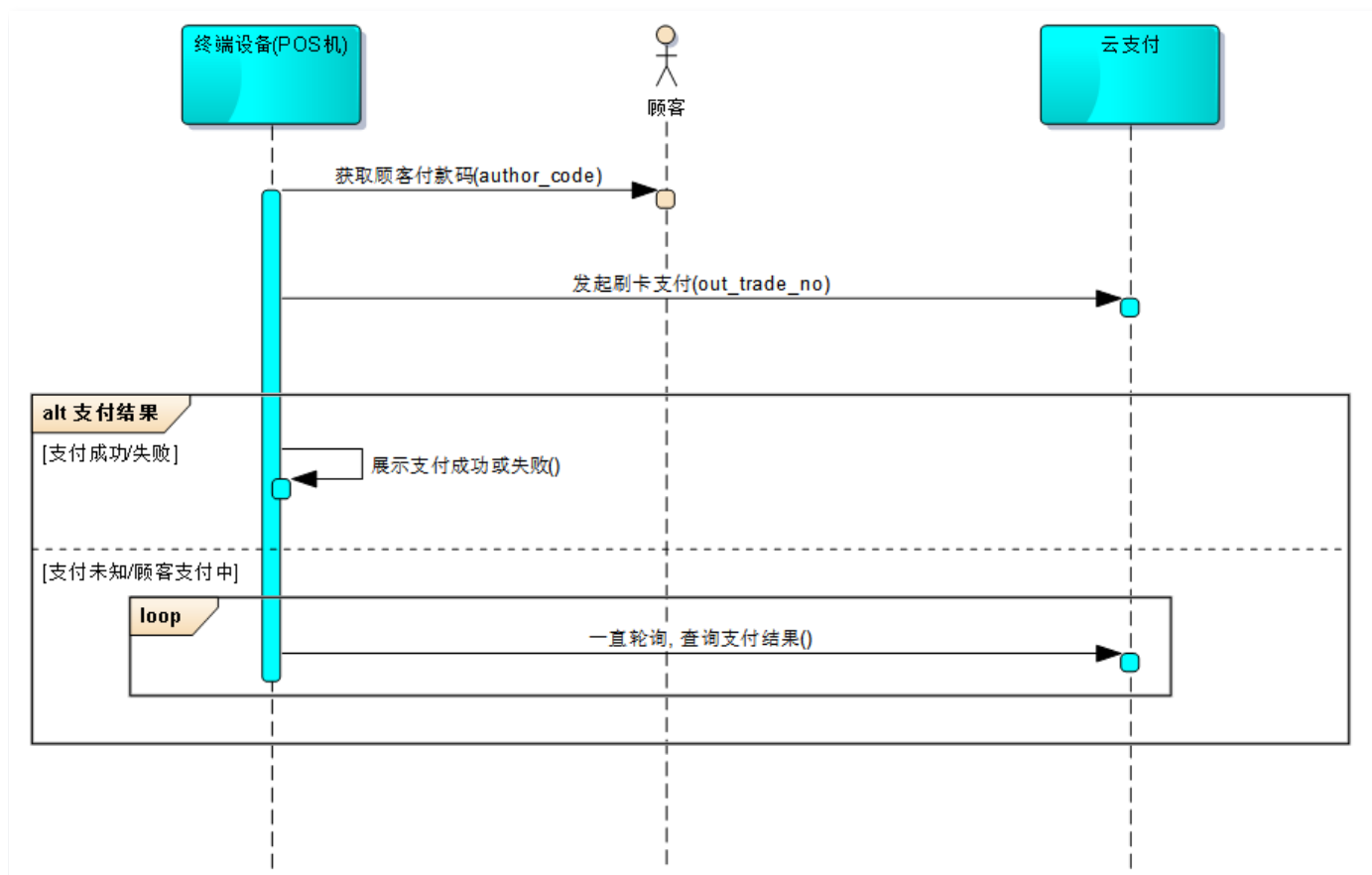
- 轮询接口: get_bill_device_bind_info, 详见 [接口说明](#)。
- 轮询时间间隔: 由 generate_bill_device_bind_qr_code 接口返回的 interval 指定, 单位为秒。假设 interval 为1, 则机具每次调用 get_bill_device_bind_info 接口完成后等待1秒再继续调用。
- 轮询结束条件有两个:
 - get_bill_device_bind_info 接口返回了绑定信息, 即应答中 activated 为 true;
 - 轮询超过了二维码有效期, 有效期长度由 generate_bill_device_bind_qr_code 接口应答中的 ttl 指定, 单位为秒, 假如 ttl 为180, 表示整个轮询过程超过180秒就必须结束, 并停止展示二维码。

8. 处理绑定信息

获取到绑定信息后 (即 activated 为 true), 把应答中的 e_authen_key 内容先做 base64 解码, 再执行解密 (解密说明详见 [10.1 AES-128-CBC 解密说明](#)), 解密密钥为机具初始密钥。解密成功后, 把解密后的内容和其它绑定信息使用机具自有加密方式加密一次, 最后写入磁盘。

8. 机具收款相关流程说明

8.1 支付

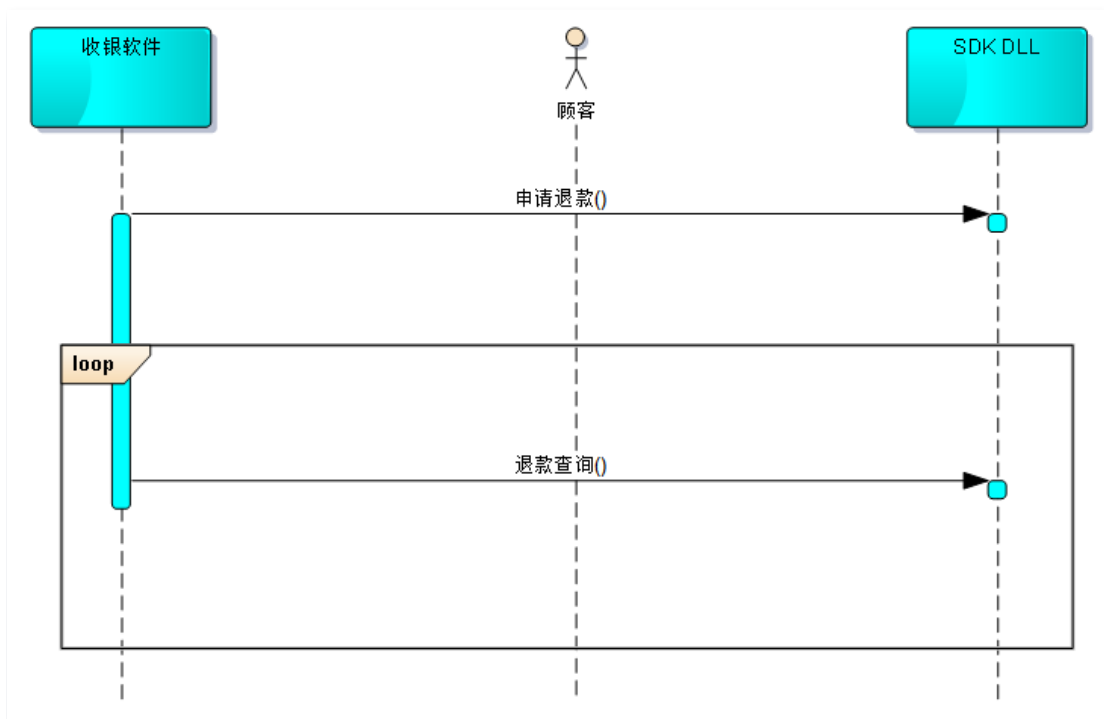


刷卡支付接口: https://pay.qcloud.com/cpay/brief_micro_pay, 详见 [9.6 刷卡支付](#)。

查询订单接口: https://pay.qcloud.com/cpay/brief_query_order, 详见 [9.7 查询支付单](#)。

- 支付方式：微信支付、支付宝支付（配置方法详见 [支付宝子商户配置](#)）、会员卡支付，云支付后台通过顾客付款码区分顾客的支付方式。
- 支付未知：网络请求超时和业务上的结果未知，详见 [9.11 其它消息体说明](#) 中 Status 的说明。
- 接口返回 status 为 0，只表示业务请求成功。订单的支付结果需通过订单的状态（即应答中的 cts 字段）来判断。
- 每次接口调用成功后等待 2 秒再查单，如果 2 分钟内查询不到结果，请到手机端管理系统查看交易明细确认支付结果。
- 支付流程完成后需要上报支付相关信息，详见 [9.10 支付信息上报](#)。

8.2 退款



退款接口：<https://pay.qqcloud.com/cpay/refund>，详见 [9.8 退款](#)。

查询退款订单接口：https://pay.qqcloud.com/cpay/query_refund_order，详见 [9.9 查询退款单](#)。

- 申请退款成功，并不代表退款成功。
- 退款是一个异步过程，申请退款成功后，只是代表第三方支付平台受理这笔退款，是否到账需要看这笔钱退到哪里。
- 如果是退款至微信零钱账户或支付宝账户，会很快到账。
- 如果是退款至顾客的银行卡，到账时间与银行处理进度相关，可能要花几个小时。
- 申请退款成功后，需告知顾客，申请退款已经成功，请关注后续到账情况。顾客可关注微信或支付宝消息通知，收到已经发起退款或退款到账的消息。
- 如果想在设备上看这笔退款是否到账，可以调用退款查询接口，通过退款单的状态来查看是否退款成功。适配者可以自己选择是否适配退款查询接口。

8.3 机具支付使用 HTTPS 长连接

1. 背景

机具与云支付后台使用 HTTPS 方式交互，如果机具收款时每次都新建连接请求云支付后台，则完成 TCP 握手和 TLS 握手需要很多次网络报文交互，会增加支付相关接口调用的整体耗时，影响收款体验。因此机具需要与云支付后台保持一个 HTTPS 长连接，使用长连接完成支付相关接口调用。

2. 创建连接

- 如果机具已绑定，机具启动完成后立即尝试创建与云支付后台的 HTTPS 连接，并保持这个连接，供后续支付使用。如果创建连接过程超时，则跳过此步骤，让机具完成启动。
- 如果机具未绑定，待机具绑定成功后，立即尝试创建与云支付后台的 HTTPS 连接，并保持这个连接，供后续支付使用。

3. 保持连接

每个 HTTPS 请求头设置 `connection: keep-alive`。每隔 2 分钟，进行一次真实的 HTTPS 交互，发送请求：

GET / HTTP/1.1\r\nHost: pay.qcloud.com\r\nConnection: keep-alive\r\n\r\n，接收完服务器应答后丢掉不处理。

4. 预热连接

商家输入完金额并按下确定键之后，每隔0.5秒通过长连接以 GET 方式向后台发送'\r\n' 2个字符，扫码后停止发送，走正常支付流程。GET 请求不会触发后台返回应答。伪代码详细说明如下：

```
function uplink_warm_up() {
  while true {
    if qrcode_is_ready {
      return
    }
    t = now_time()

    https_send('\r\n', 2)
    wait_ack() // 不同机具可能做法不同
    while now_time() - t < 500ms {
      sleep(50ms)
    }
  }
}
```

9. 接口说明

9.1 接口规范

摘要	详细说明
传输方式	HTTPS
提交方式	POST
字符编码	UTF-8
内容格式	content_type: application/json
证书校验	严格校验服务端证书。
应答校验	如果应答设置了 authen_info（或 a）则严格校验 authen_code（或 a）的值。

9.2 获取云支付后台系统时间

接口地址

https://pay.qcloud.com/cpay/ping

请求参数

POST 空字符串即可，后台不校验 POST 内容。

应答参数

参数名	必填	类型	长度(Byte)	说明
timestamp	是	Int	8	后台系统的 UNIX 时间戳，单位为秒，如1568810504。
status	是	Int	4	错误码。
description	是	String	最长128	status 对应的错误信息。

示例说明

```
curl -H "Content-Type:application/json" -X POST -data "<https://pay.qcloud.com/cpay/ping>
{"description":"ok","status":0,"timestamp":1568810077}
```

9.3 检查机具绑定信息

接口地址

https://pay.qcloud.com/cpay/check_bill_device_bind_info

请求参数

参数名	必填	类型	长度	说明
request_content	是	String	—	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	—	认证信息，详见“初始密钥认证信息”。

RequestContent 结构

参数名	必填	类型	长度 (Byte)	说明
sn_code	是	String	—	机具自身的 sn，注意要是完整的 sn，如 LANDI_QM800_LA19*****，POPSECU_SL51_BP51*****。
out_shop_id	是	String	—	云支付系统内的全局门店 ID，机具绑定成功后即可获得这个 ID。
device_id	是	String	—	云支付系统内的逻辑设备 ID，机具绑定成功后即可获得这个 ID。
nonce_str	是	String	8	随机字符串，ASCII 字符（0-9、a-z、A-Z）。
hash_content	是	String	64	authen_key（支付密钥）和随机字符串一起计算的 HMAC-SHA256 的值，authen_key 是在机具绑定成功后解密 e_authen_key 字段获得。
timestamp	是	Int	8	机具系统当前时间，UNIX 时间戳，单位为秒，如1568810504。

应答参数

参数名	必填	类型	长度(Byte)	说明
response_content	是	ResponseContent	—	应答内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	—	认证信息，详见“初始密钥认证信息”。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
status	是	Int	4	错误码。
description	是	String	最长128	错误信息。
log_id	是	Int	4	消息 ID。
internal_status	是	Int	4	具体说明见 internal_status 错误码表。

示例说明

- 请求生成示例代码：

```
std::string nonce_str = "ea90ceba" // 生成8字节随机字符串
std::string authen_key_hash = hmac_sha256(authen_key, nonce_str); // 计算 authen_key 的 hash 值

Json::Value request_content;
request_content["sn_code"] = "aaa";
request_content["out_shop_id"] = "ddd";
request_content["device_id"] = "fff";
request_content["nonce_str"] = nonce_str;
request_content["hash_content"] = authen_key_hash;
request_content["timestamp"] = 1568859333; //当前系统时间

Json::FastWriter w;
std::string request_content_str = w.write(request_content);
std::string authen_code = hmac_sha256(init_key, request_content_str); // 使用机具初始密钥计算签名

Json::Value authen;
authen["authen_code"] = authen_code
authen["authen_type"] = 1; //hmac_sha256 为 1

Json::Value authen_info;
authen_info["a"] = authen; //认证码

Json::Value request; //构造最终发给服务器的请求
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;

std::string request_str = w.write(request);
```

request_str 即为 post 内容。

- 请求内容示例：

```
{
  "authen_info":{
    "a":{
      "authen_code":"091745D976208923097F66771324CA7C6EC32A55ABECAD6C3D42826AA363CA86",
      "authen_type":1
    }
  },
  "request_content":{"
    "device_id":"fff",
    "hash_content":"2F09EFCB064D4F9A4A0BABD5260842D8AC4D0611F4B941A03A77E5E55B5168D7",
    "nonce_str":"ea90ceba",
    "out_shop_id":"ddd",
    "sn_code":"aaa",
    "timestamp":1568859333
  }"
}
```

- 应答内容示例：

```
{
  "response_content":{"
    "status":0,
    "description": "",
```

```

"log_id":18654852,
"internal_status":0,
},
"authen_info":{
  "a":{
    "authen_type":1,
    "authen_code":"xxxx"
  }
}
}

```

9.4. 获取绑定二维码相关信息

接口地址

https://pay.qcloud.com/cpay/generate_bill_device_bind_qr_code

请求参数

参数名	必填	类型	长度	说明
request_content	是	String	-	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	-	认证信息，详见“初始密钥认证信息”。

RequestContent 结构

参数名	必填	类型	长度 (Byte)	说明
sn_code	是	String	-	机具自身的 sn，注意要是完整的 sn，如 LANDI_QM800_LA19*****，POPSECU_SL51_BP51*****。
mcc	否	String	-	基站信息中的 mcc 字段，如果未获取到可以不填。
mnc	否	String	-	基站信息中的 mnc 字段，如果未获取到可以不填。
lac	否	String	-	基站信息中的 lac 字段，如果未获取到可以不填。
cellid	否	String	-	基站信息中的 cellid 字段，如果未获取到可以不填。
rss	否	String	-	基站信息中的 rss 字段，如果未获取到可以不填。
app_cpay_version	是	String	-	由云支付指定的版本号，初始值为1.0.0，需要编译到机具代码中，每次机具升级程序需要向云支付申请一个 app_cpay_version。
nonce_str	是	String	8	随机字符串，ASCII 字符（0-9、a-z、A-Z）。
timestamp	是	Int	8	机具系统当前时间，UNIX 时间戳，单位为秒，如1568810504。

应答参数

参数名	必填	类型	长度(Byte)	说明
-----	----	----	----------	----

response_content	是	ResponseContent	-	应答内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	-	认证信息，详见“初始密钥认证信息”。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
status	是	Int	4	错误码。
description	是	String	最长128	错误信息。
log_id	是	Int	4	消息 ID。
internal_status	是	Int	4	具体说明见 internal_status 错误码表。
generate_bill_device_bind_qr_code_response	否	String	-	只有 status 为0的时候才存在，具体见 GenerateBillDeviceBindQrCodeResponse 说明。

GenerateBillDeviceBindQrCodeResponse

参数名	必填	类型	长度(Byte)	说明
activator	是	String	12	二维码唯一标识，用于生成绑定二维码。
token	是	String	12	轮询绑定结果时要用到的票据。
ttl	是	Int	4	二维码有效时间，单位为秒，如180，表示二维码3分钟内有效。
mini_program_url	是	Int	4	收款设备小程序的 url，用于生成绑定二维码。
interval	是	Int	4	轮询绑定结果的时间间隔，单位为秒。

示例说明

- 请求生成示例代码：

```

Json::Value request_content;
request_content["sn_code"] = "aaa";
request_content["mcc"] = "bbb";
request_content["mnc"] = "ccc";
request_content["lac"] = "ddd";
request_content["cellid"] = "eee";
request_content["rss"] = "fff";
request_content["app_cpay_version"] = "1.0.0";
request_content["nonce_str"] = "ea90ceba"; // 8字节随机字符串
request_content["timestamp"] = 1568859333; // 当前系统时间

Json::FastWriter w;
std::string request_content_str = w.write(request_content);
std::string authen_code = hmac_sha256(init_key, request_content_str); // 使用机具初始密钥计算签名

Json::Value authen;
authen["authen_code"] = authen_code;
authen["authen_type"] = 1; //hmac_sha256 为 1

Json::Value authen_info;
authen_info["a"] = authen; //认证码，签名是 s

```

```
Json::Value request; //构造最终发给服务器的请求
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;

std::string request_str = w.write(request);
```

request_str 即为 post 内容。

- 请求内容示例:

```
{
  "authen_info":{
    "a":{
      "authen_code":"E84DCDFD546D0A0F91FFB90746F6FC7E9A8FCF62C4E682FE769BFB64486F72E5",
      "authen_type":1
    }
  },
  "request_content":{"
    "app_cpay_version":"1.0.0",
    "cellid":"eee",
    "lac":"ddd",
    "mcc":"bbb",
    "mnc":"ccc",
    "nonce_str":"ea90ceba",
    "rss":"fff",
    "sn_code":"aaa",
    "timestamp":1568859333
  }"
}
```

- 应答内容示例:

```
{
  "response_content":{"
    "status":0,
    "description":"",
    "log_id":18654852,
    "internal_status":0,
    "generate_bill_device_bind_qr_code":{"
      "activator":"p9fa01zex4mi",
      "token":"8azip8115bq",
      "ttl":180,
      "mini_program_url":"https://payapp.weixin.qq.com/smartqr/entry/home",
      "interval":1
    }"
  },
  "authen_info":{
    "a":{
      "authen_type":1,
      "authen_code":"xxxx"
    }
  }
}
```

9.5 获取绑定信息

接口地址

https://pay.qcloud.com/cpay/get_bill_device_bind_info

请求参数

参数名	必填	类型	长度	说明
request_content	是	String	–	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	AuthenInfo	–	认证信息，详见“初始密钥认证信息”。

RequestContent 结构

参数名	必填	类型	长度(Byte)	说明
sn_code	是	String	–	机具自身的 sn，注意要是完整的 sn，如 LANDI_QM800_LA19*****，POPSECU_SL51_BP51*****。
token	是	String	12	generate_bill_device_bind_qr_code 接口应答中的 token。
nonce_str	是	String	8	随机字符串，ASCII 字符（0–9、a–z、A–Z）。
timestamp	是	Int	8	机具系统当前时间，UNIX 时间戳，单位为秒，如1568810504。

应答参数

参数名	必填	类型	长度(Byte)	说明
response_content	是	ResponseContent	–	应答内容，详见本节 ResponseContent。
authen_info	否	AuthenInfo	–	认证信息，详见“初始密钥认证信息”。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
status	是	Int	4	错误码。
description	是	String	最长128	错误信息。
log_id	是	Int	4	消息 ID。
internal_status	是	Int	4	具体说明见 internal_status 错误码表。
get_bill_device_bind_info_response	否	String	–	只有 status 为0的时候才存在，具体见 GetBillDeviceBindInfoResponse 说明。

GetBillDeviceBindInfoResponse

参数名	必填	类型	长度(Byte)	说明
activated	是	Int	4	后台是否绑定完成，0–未完成，1–已完成。
out_mch_id	否	String	最长64	云支付内部的服务商 ID，只当 activated 为1时才有。
out_sub_mch_id	否	String	最长64	云支付内部的子商户 ID，只当 activated 为1时才有。
cloud_cashier_id	否	String	最长64	云支付规定的定单前缀，只当 activated 为1时才有。

authen_type	否	Int	4	子商户与云支付之间的签名算法类型，只当 activated 为1时才有。
e_authen_key	否	String	最长256	Aes128CBC 加密并 base64 编码之后的支付密钥，机具先 base64 解码后，再使用初始化密钥来解密这个字段，以获得支付密钥（authen_key）。支付用这个密钥来计算签名。只当 activated 为1时才有。
out_shop_id	否	String	最长64	云支付系统全局唯一门店 ID，只当 activated 为1时才有。
shop_name	否	String	最长128	门店名称，只当 activated 为1时才有。
device_id	否	String	最长64	设备 ID，只当 activated 为1时才有。
device_name	否	String	最长256	设备名称，只当 activated 为1时才有。
staff_id	否	String	最长64	店员 ID，保留字段，暂未使用。
staff_name	否	String	最长256	店员名称，保留字段，暂未使用。
device_shift_type	否	Int	4	设备类型，只当 activated 为1时才有。
nonce_str	是	String	32	随机字符串，ASCII 字符（0-9、a-z、A-Z）。

示例说明

- 请求生成示例代码：

```
Json::Value request_content;
request_content["sn_code"] = "aaa";
request_content["token"] = "8azip8115bq";
request_content["nonce_str"] = "ea90ceba"; // 8字节随机字符串
request_content["timestamp"] = 1568859333; // 当前系统时间

Json::FastWriter w;
std::string request_content_str = w.write(request_content);
std::string authen_code = hmac_sha256(init_key, request_content_str); // 使用机具初始密钥计算签名

Json::Value authen;
authen["authen_code"] = authen_code
authen["authen_type"] = 1; //hmac_sha256 为 1

Json::Value authen_info;
authen_info["a"] = authen; //认证码，签名是 s

Json::Value request; //构造最终发给服务器的请求
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;

std::string request_str = w.write(request);
```

request_str 即为 post 内容。

- 请求内容示例：

```
{
  "authen_info":{
    "a":{
      "authen_code":"09E33D5E1D208569EE634C733F0B1FD933DD690B25F5BF72B386257B7A2113B0",
      "authen_type":1
    }
  },
  "request_content":{
```

```
"nonce_str":"ea90ceba",
"sn_code":"aaa",
"timestamp":1568859333,
"token":"8azip8115bq"
}"
}
```

● 应答内容示例:

```
{
  "response_content": "{
    "status":0,
    "description":"",
    "log_id":18654852,
    "internal_status":0,
    "bill_device_bind_sn" : "{
      "activated":1,
      "out_mch_id":"aaa",
      "out_sub_mch_id":"bbb",
      "cloud_cashier_id":"ccc",
      "authen_type":1,
      "authen_key":"ddd",
      "out_shop_id":"eee",
      "shop_name":"fff",
      "device_id":"ggg",
      "device_name":"hhh",
      "device_shift_type":1,
      "nonce_str":"416492026bc84091bcaf7e74ea90ceba"
    }"
  }",
  "authen_info":{
    "a":{
      "authen_type":1,
      "authen_code":"xxxx"
    }
  }
}
```

9.6 刷卡支付

接口地址

https://pay.qcloud.com/cpay/brief_micro_pay

请求参数

参数名	必填	类型	长度	说明
r	是	RequestContent	-	请求内容, 详见本节 RequestContent。
a	是	string	-	使用支付密钥计算的认证码, 目前只支持 sha256 计算认证码。

RequestContent 结构

参数名	必填	类型	长度(Byte)	说明
sc	是	String	-	机具的 sn, 注意要是完整的 sn, 如: LANDI_QM800_LA19***** , POPSECU_SL51_BP51*****。
osi	是	String	-	云支付分配的门店全局 ID, 由绑定成功时获取。

ns	是	String	32	随机字符串，ASCII 字符（0-9、a-z、A-Z）。
otn	是	String	-	由调用方生成的订单号，前缀必须是绑定成功时获取的云支付订单前缀。
ac	是	String	-	author_code，刷卡支付时顾客的付款码。云支付后台通过这个字段来区分是微信支付，或支付宝支付还是会员卡支付。
tf	是	Int	8	即 total_fee，订单总金额，单位分。
bd	是	String	-	即 body，商品或订单简要描述。
di	是	String	-	即 device_id，设备编号，由绑定成功时获取。
stt	是	Int	-	即 sub_terminal_type，子终端类型，代表一个机具品牌，由云支付分配，详见 5.1 申请初始资源 。

应答参数

参数名	必填	类型	长度(Byte)	说明
rc	是	ResponseContent	-	请求内容，详见本节 ResponseContent。
a	是	String	32	认证码，目前只支持 sha256 计算认证码。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
s	是	Int	4	错误码。
d	是	String	最长128	即 description，错误信息。
li	否	Int	4	即 log_id，消息 ID。
is	是	Int	4	具体说明见 internal_status 错误码表。
brief_micro_pay	否	BriefMicroPayResponse	-	详见本节 BriefMicroPayResponse。

BriefMicroPayResponse

参数名	必填	类型	长度(Byte)	说明
wx	否	BriefWxpayOrderContentExt	-	详见 BriefWxpayOrderContentExt。
ali	否	BriefAlipayOrderContentExt	-	详见 BriefAlipayOrderContentExt。
cd	否	BriefCardOrderContentExt	-	详见 BriefCardOrderContentExt。
ns	是	String	8	8字节随机字符串。

示例说明

请求生成示例代码：

```
Json::Value request_content;  
request_content["ns"] = generate_random_nonce_str();  
request_content["osi"] = out_shop_id;  
request_content["otn"] = out_trade_no;  
request_content["ac"] = author_code;  
request_content["tf"] = total_fee;  
request_content["bd"] = body;  
request_content["di"] = device_id;
```

```
request_content["sc"] = sn_code;
request_content["stt"] = sub_terminal_type;
Json::FastWriter w;
const std::string &rc = w.write(request_content);
std::string authen_code = hmac_sha256(authen_key, rc); // 计算签名
Json::Value request;
request["r"] = rc;
request["a"] = authen_code;
return w.write(request);
```

请求内容示例:

```
{
  "r": "{
    \"ns\": \"69050786d195198efe9070ae45857070\",
    \"osi\": \"sz01QgJZkxDB7RVeL6XY\",
    \"otn\": \"0100826651692015704554111\",
    \"ac\": \"135005875955002257\",
    \"tf\": 900,
    \"bd\": \"支付简述\",
    \"di\": \"tdFQtGI2M9MOcgTn\",
    \"sc\": \"LANDI_QM800_LA198KBQ8W1666\",
    \"stt\": 2451
  }",
  "a": "A4BCE4DF5C59C1D1532FA38D5C0298E46F224D02D38EFA6A766A5368271413B1"
}
```

应答内容示例:

```
{
  "rc": "{
    \"s\": 0,
    \"d\": \"xxx\",
    \"li\": 2643575008,
    \"is\": 0,
    \"brief_micro_pay\": {
      \"wx\": {
        \"cts\": 2
      },
      \"ns\": \"dJMSHgYs\"
    }
  },
  \"a\": \"4E80102BC41267241DE4B55AC40A5B2952C2F739E384767DDD065B8E1189846A\"
}
```

9.7 查询支付单

接口地址

https://pay.qcloud.com/cpay/brief_query_order

请求参数

参数名	必填	类型	长度	说明
r	是	RequestContent	—	请求内容, 详见本节 RequestContent。
a	是	string	—	使用支付密钥计算的认证码, 目前只支持 hmac-sha256 计算认证码。

RequestContent 结构

参数名	必填	类型	长度(Byte)	说明
osi	是	String	–	云支付分配的门店全局 ID，由绑定成功时获取。
ns	是	String	32	随机字符串，ASCII 字符（0-9、a-z、A-Z）。
otn	是	String	–	由调用方生成的支付订单号。

应答参数

参数名	必填	类型	长度(Byte)	说明
rc	是	ResponseContent	–	请求内容，详见本节 ResponseContent。
a	是	String	32	使用支付密钥计算的认证码，目前只支持 hmac-sha256 计算认证码。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
s	是	Int	4	错误码。
d	是	String	最长128	即 description，错误信息。
li	否	Int	4	即 log_id，消息 ID。
is	是	Int	4	具体说明见 internal_status 错误码表。
brief_query_order	否	BriefQueryOrderResponse	–	详见本节 BriefQueryOrderResponse。

BriefQueryOrderResponse

参数名	必填	类型	长度(Byte)	说明
pp	是	Int	4	第三方支付平台，详细见 PayPlatform。
otn	是	String	–	由支付调用户生成的订单号。
tf	是	Int	8	订单总金额，单位为分。
wx	否	BriefWxpayOrderContentExt	–	详见 BriefWxpayOrderContentExt。
ali	否	BriefAlipayOrderContentExt	–	详见 BriefAlipayOrderContentExt。
cd	否	BriefCardOrderContentExt	–	详见 BriefCardOrderContentExt。
ns	是	String	8	8字节随机字符串。

示例说明

● 请求生成示例代码：

```
Json::Value request_content;
request_content["ns"] = generate_random_nonce_str();
request_content["osi"] = out_shop_id;
request_content["di"] = device_id;
request_content["sc"] = sn_code;
request_content["otn"] = out_trade_no;

Json::FastWriter w;
```

```
const std::string &rc = w.write(request_content);

std::string authen_code = hmac_sha256(authen_key, rc); // 计算签名

Json::Value request;
request["r"] = rc;
request["a"] = authen_code;
return w.write(request);
```

● 请求内容示例:

```
{
  "r":{"
    "ns":"92f7ba8154a4532d97cb6f2b181309e9",
    "otn":"013907131297815704553502",
    "osi":"sz015Xco0HsfMcFGkeB9"
  },
  "a":"D9C1233C5D08288A00975F61DF328E7E76F426C3519284A1F43245703F7CFCFE"
}
```

● 应答内容示例:

```
{
  "rc":{"
    "s":0,
    "d":"u64CDu4F5Cu6210u529Fu3002",
    "li":2185455507,
    "is":0,
    "brief_query_order":{"
      "otn":"013907131297815704553502",
      "tf":240000,
      "pp":2,
      "ali":{"
        "cts":4
      },
      "ns":"sclBgLM9"
    },
    "a":"C8A4986C17BAB6F88623CBA0F89C375288F972C43181E193A22EC6BDA25CFDAA"
  },
  "a":"C8A4986C17BAB6F88623CBA0F89C375288F972C43181E193A22EC6BDA25CFDAA"
}
```

9.8 退款

接口地址

<https://pay.qcloud.com/cpay/refund>

说明文档链接

[申请退款](#)

9.9 查询退款单

接口地址

https://pay.qcloud.com/cpay/query_refund_order

说明文档链接

[查询退款单](#)

9.10 支付信息上报

接口地址

https://pay.qcloud.com/cpay/device_report

请求参数

参数名	必填	类型	长度	说明
request_content	是	RequestContent	-	请求内容，该 string 可以转为 json 结构，json 格式见本节 RequestContent。
authen_info	是	Authen_info	-	认证信息，详见“支付密钥认证信息”。

RequestContent 结构

参数名	必填	类型	长度 (Byte)	说明
device_type	是	Int	4	设备类型：1-移动收款机具。
company_code	是	String	-	公司简码，4.1节中有说明。
model	是	String	-	型号简码，4.1节中有说明。
sn	是	String	-	完整 sn，4.1节中有说明。
begin_timestamp	是	Int	8	统计开始时间，UNIX 时间戳，单位为秒，如 1568810504。
end_timestamp	是	Int	8	统计结束时间，UNIX 时间戳，单位为秒，如 1568810504。
sim_number	否	String	-	Sim 卡号，即 ICCID 号。
cpu_usage	否	Int	4	当前 CPU 使用率0-100。
mem_max	否	Int	4	最大内存 KB。
mem_in_use	否	Int	4	当前已用内存 KB。
disk_max	否	Int	4	最大硬盘 KB。
disk_in_use	否	Int	4	当前已用硬盘 KB。
network_type	是	String	-	支付所使用的网络类型：wifi、2g、3g、4g 等。
upload_net_flow	否	Int	4	累计上传流量。
download_net_flow	否	Int	4	累计下载流量。
mcc	否	String	-	基站国家码，十进制数字字符串。
mnc	否	String	-	基站网络码，十进制数字字符串。
lac	否	String	-	基站小区号，十进制数字字符串。
cellid	否	String	-	基站 ID，十进制数字字符串。
rss	否	String	-	基站信号强度，单位 dbm，十进制数字字符串。
longitude	否	String	-	经度。

latitude	否	String	-	纬度。
wx_pay_count	是	Int	4	累计微信支付笔数，上报后清零计数。
wx_pay_success_count	是	Int	4	累计微信支付成功笔数，上报后清零计数。
ali_pay_count	是	Int	4	累计支付宝笔数，上报后清零计数。
ali_pay_success_count	是	Int	4	累计支付宝成功笔数，上报后清零计数。
other_pay_count	是	Int	4	累计其他支付笔数，上报后清零计数。
other_pay_success_count	是	Int	4	累计其他支付成功笔数，上报后清零计数。
wx_pay_amount	是	Int	8	累计微信支付金额，单位分，上报后清零计数。
ali_pay_amount	是	Int	8	累计支付宝金额，上报后清零计数。
other_pay_amount	是	Int	8	累计其他支付金额，上报后清零计数。
out_mch_id	是	String	-	云支付服务商号，绑定成功时获取。
out_sub_mch_id	是	String	-	云支付子商户号，绑定成功时获取。
cloud_cashier_id	是	String	-	云支付分配给商户订单前缀，绑定成功时获取。
authen_type	是	Int	4	签名类型1: HMAC-SHA256，绑定成功获取。
out_shop_id	是	String	-	云支付门店全局 ID，绑定成功时获取。
device_id	是	String	-	云支付设备 ID，绑定成功时获取。
staff_id	否	String	-	云支付店员 ID，绑定成功时获取，有可能没有。
app_cpay_version	是	String	-	云支付统一分配的版本号。
api_calling_infos	否	ApiCallingInfo	-	接口调用数据，见 ApiCallingInfo。

ApiCallingInfo 结构

参数名	必填	类型	长度 (Byte)	说明
out_trade_no	是	String	-	支付时的订单号。
code	是	Int	4	如果能正常获取 HTTP 的状态码（如200、404、405）则用状态码填充，其它情况：连接失败填（-2），接口获取应答超时填（-3），未知问题填（-4）。
status	是	Int	4	brief_micro_pay 应答的一级错误码，应答中 s 字段。
internal_status	是	Int	4	brief_micro_pay 应答的二级错误码，应答中 is 字段。
delay_ms	是	Int	4	从扫用户收款码开始到获取到 brief_micro_pay 接口应答中间的耗时，单位为毫秒。
description	是	String	-	brief_micro_pay 应答中的 d 字段内容。
pay_platform	是	Int	4	支付方式，见 PayPlatform 说明。
order_status	否	Int	4	支付完成最后一次云支付接口（可能是 brief_micro_pay 或

				brief_query_order) 返回的订单状态, 即 cts 字段内容。
trade_state_desc	否	String	-	支付完成最后一次云支付接口 (可能是 brief_micro_pay 或 brief_query_order) 返回的订单状态描述, 即 tsd 字段内容。

应答参数

参数名	必填	类型	长度(Byte)	说明
response_content	是	ResponseContent	-	应答内容, 详见本节 ResponseContent。
authen_info	否	AuthenInfo	-	认证信息, 详见“支付密钥认证信息”。

ResponseContent 结构

参数名	必填	类型	长度(Byte)	说明
status	是	Int	4	错误码。
description	是	String	最长128	错误信息。
log_id	是	Int	4	消息 ID。
internal_status	是	Int	4	具体说明见 internal_status 错误码表。

示例说明

- 请求生成示例代码:

```
Json::Value request_content;  
request_content["device_type"] = 1;  
request_content["company_code"] = "aa";  
request_content["model"] = "bb";  
request_content["sn"] = "cc";  
request_content["sim_number"] = "dd";  
request_content["cpu_usage"] = 33;  
request_content["mem_max"] = 100000;  
request_content["mem_in_use"] = 2000;  
request_content["disk_max"] = 200000;  
request_content["disk_in_use"] = 39999;  
request_content["network_type"] = "ccc";  
request_content["upload_net_flow"] = 33333;  
request_content["download_net_flow"] = 4444444;  
request_content["mcc"] = "fff";  
request_content["mnc"] = "ggg";  
request_content["lac"] = "aaa";  
request_content["cellid"] = "bbb";  
request_content["rss"] = "ccc";  
request_content["longitude"] = "ddd";  
request_content["latitude"] = "eee";  
request_content["begin_timestamp"] = 1553268531;  
request_content["end_timestamp"] = 1553268531;  
request_content["wx_pay_count"] = 11;  
request_content["wx_pay_success_count"] = 11;  
request_content["ali_pay_count"] = 12;  
request_content["ali_pay_success_count"] = 12;  
request_content["other_pay_count"] = 13;  
request_content["other_pay_success_count"] = 13;  
request_content["wx_pay_amount"] = 22;  
request_content["ali_pay_amount"] = 23;  
request_content["other_pay_amount"] = 24;
```

```

request_content["out_mch_id"] = "gggg";
request_content["out_sub_mch_id"] = "a1";
request_content["cloud_cashier_id"] = "b1";
request_content["authen_type"] = 1;
request_content["out_shop_id"] = "d1";
request_content["device_id"] = "e1";
request_content["staff_id"] = "f1";
request_content["device_shift_type"] = "g1";
request_content["nonce_str"] = "416492026bc84091bcaf7e74ea90ceba";

Json::Value info_0;
info_0["out_trade_no"] = "EEEEEEEEEEEEEE";
info_0["code"] = 200;
info_0["status"] = 2;
info_0["internal_status"] = 2;
info_0["delay_ms"] = 200;
info_0["description"] = "操作成功";
info_0["pay_platform"] = 1;
info_0["order_status"] = 3;
info_0["trade_state_desc"] = "支付成功";

request_content["api_calling_infos"].append(info_0);

Json::Value info_1;
info_1["out_trade_no"] = "FFFFFFF";
info_1["code"] = 200;
info_1["status"] = 22;
info_1["internal_status"] = 22;
info_1["delay_ms"] = 300;
info_1["description"] = "xxx";
info_1["pay_platform"] = 1;
info_1["order_status"] = 3;
info_1["trade_state_desc"] = "xxx";

request_content["api_calling_infos"].append(info_1);

Json::FastWriter w;
std::string request_content_str = w.write(request_content);
std::string authen_code = hmac_sha256(authen_key, request_content_str); // 计算签名

Json::Value authen;
authen["authen_code"] = authen_code;
authen["authen_type"] = 1; //统一使用 hmac_sha256, 填1

Json::Value authen_info;
authen_info["a"] = authen; //key 填"a"就行, 表示认证码

Json::Value request;
request["request_content"] = request_content_str;
request["authen_info"] = authen_info;
std::string request_str = w.write(request);

```

request_str 即为 post 内容。

- 请求内容示例:

```

{
  "request_content": "{
    "nonce_str": "4db12671e9510667d9c8d65bdf50a27b",
    "company_code": "LANDI",

```

```

"model": "LANDI_QM600",
"sn": "LANDI_QM600_LA197LB331212",
"begin_timestamp": 1570458756,
"end_timestamp": 1570462751,
"network_type": "wifi",
" MCC": "460",
" MNC": "00",
" LAC": "2733",
" cellid": "3437",
" rssi": "-79",
"sim_number": "89860412101871062566",
"wx_pay_count": 1,
"wx_pay_success_count": 1,
"ali_pay_count": 0,
"ali_pay_success_count": 0,
"ali_pay_amount": 0,
"other_pay_count": 0,
"other_pay_success_count": 0,
"other_pay_amount": 0,
"out_mch_id": "sz01M43ouf007JL1sn33",
"out_sub_mch_id": "sz01NI6PoQVvYEmOcoRp",
"cloud_cashier_id": "010110893",
"authen_type": 1,
"out_shop_id": "sz01aZk73lnFVxX6XSWp",
"device_id": "kUwbvm1cpdGn4Kxm",
"device_type": 1,
"device_shift_type": 1,
"app_cpay_version": "1.0.0",
"api_calling_infos": [
  {
    "out_trade_no": "0101108936210615704627481",
    "code": 200,
    "status": 0,
    "internal_status": 0,
    "delay_ms": 2060
  }
],
},
"authen_info": {
  "a": {
    "authen_type": 1,
    "authen_code": "C80A72B0A7858D2D84E26093F562516C64DB1D319534A7D1B9ACF17556226019"
  }
}
}

```

● 应答内容示例:

```

{
  "response_content": {
    "status": 0,
    "description": "xxxx",
    "log_id": 973138369,
    "internal_status": 0,
    "device_info_report": {
      "nonce_str": "4dOOZyhxzTRexn5OBDqA8vhve3FqnCHL"
    }
  },
  "authen_info": {

```

```
"a":{
  "authen_type":1,
  "authen_code":"5E417C5D6FE523982C3507A22D762F3215255F752C3CD4DC37C7792D964418C3"
}
```

9.11 查询订单明细

接口地址

https://pay.qcloud.com/cpay/brief_query_order_list

请求参数

参数名	必填	类型	长度	说明
r	是	RequestContent	-	请求内容，详见本节 RequestContent。
a	是	string	-	使用支付密钥计算的认证码，目前只支持 hmac-sha256 计算认证码。

RequestContent 结构

参数名	必填	类型	长度 (Byte)	说明
spps	否	Int []	单个元素 4	sub_pay_platforms: 子支付平台列表，100: 普通微信支付，200: 普通支付宝，300: 会员卡，机具填写这三个值即可。
osmi	是	String	-	out_sub_mch_id: 云支付分配的商户 ID，绑定成功后可获取此 ID。
osi	是	String	-	out_shop_id: 云支付分配的门店 ID，绑定成功后可获取此 ID。
di	是	String	-	device_id: 云支付分配的设备ID，绑定后可获取此 ID。
ot	是	Int	4	order_type: 单据类型，1: 支付订单，2: 退款单，3: 全部单据。
st	是	Int	8	start_time: 查询开始时间。
et	是	Int	8	end_time: 查询结束时间。
pn	是	Int	4	page_num: 页码，必需大于0。
ps	是	Int	4	page_size: 单页条数，必需大于0。
tt	否	Int	4	交易类型，1: 刷卡支付，2: 扫码支付，3: 公众号支付，4: App 支付，5: 声波支付，6: H5 支付，8: 一码付支付，9: 小程序支付。机具应填1。
ns	是	String	32	随机字符串，ASCII 字符（0-9、a-z、A-Z）。

应答参数

参数名	必填	类型	长度(Byte)	说明
rc	是	ResponseContent	-	请求内容，详见本节 ResponseContent。
a	是	String	32	使用支付密钥计算的认证码，目前只支持 hmac-sha256 计算认证码。

ResponseContent 结构

参数名	必填	类型	长度 (Byte)	说明

s	是	Int	4	错误码。
d	是	String	最长128	即 description，错误信息。
li	否	Int	4	即 log_id，消息 ID。
is	是	Int	4	具体说明见 internal_status 错误码表。
brief_query_order_list	否	BriefQueryOrderListResponse	-	详见本节 BriefQueryOrderListResponse。

BriefQueryOrderResponse

参数名	必填	类型	长度(Byte)	说明
tc	是	Int	4	total_count：订单总数。
ods	否	OrderDetail []	-	order_details：见 OrderDetail 说明。
ns	是	String	32	随机字符串，ASCII 字符（0-9、a-z、A-Z）。

OrderDetail

参数名	必填	类型	长度(Byte)	说明
bo	否	BriefOrder	-	订单结构体，见本节 BriefOrder。
bro	否	BriefRefundOrder	-	退款单结构体，见本节 BriefRefundOrder。

order、refund_order 只包含一个。

BriefOrder

参数名	必填	类型	长度(Byte)	说明
spp	否	Int	4	sub_pay_platform：子支付平台，100：普通微信支付，200：普通支付宝，300：会员卡。
tt	否	Int	4	trade_type：交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
otn	是	String	32	out_trade_no：支付单号。
tf	否	Int	8	total_fee：订单总金额。
ct	是	Int	8	create_time：订单创建时间。
wx_cts	否	Int	4	wxpay_current_trade_state：见枚举值定义 WxpayOrderState。
ali_cts	否	Int	4	支付宝订单状态，见枚举值定义 AlipayOrderState。
card_cts	否	int	4	card_current_trade_state：会员卡订单状态，见枚举值定义 CardOrderState。

BriefRefundOrder

参数名	必填	类型	长度(Byte)	说明
spp	否	Int	4	sub_pay_platform：子支付平台，100：普通微信支付，200：普通支付宝，300：会员卡。
tt	否	Int	4	trade_type：交易类型，1：刷卡支付，2：扫码支付，3：公众号支付，4：App 支付，

				5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
otn	是	String	32	out_trade_no：支付单号。
tf	是	Int	8	total_fee：订单总金额。
orn	是	String	32	out_refund_no：退款单号。
rf	是	Int	8	refund_fee：本次退款总金额。
ct	是	Int	8	create_time：退款单创建时间。
wx_rs	否	Int	4	wxpay_refund_state：微信退款状态，见枚举值定义 WxpayRefundOrderState。
ali_rs	否	Int	4	alipay_refund_state：支付宝退款状态，见枚举值定义 AlipayRefundOrderState。
card_rs	否	int	4	card_refund_state：会员卡退款状态，见枚举值定义 CardRefundOrderState。

示例说明

- 请求生成示例代码：

```

Json::Value request_content;
request_content["spps"].append(100);
request_content["spps"].append(200);
request_content["osmi"] = "sz0138zjfiwezee";
request_content["osi"] = "sz01FzZfei91";
request_content["di"] = "FzZfei91iefaefz";
request_content["ot"] = 1;
request_content["st"] = 1571817655;
request_content["et"] = 1571817755;
request_content["pn"] = 1;
request_content["ps"] = 2;
request_content["tt"] = 1;
request_content["ns"] = generate_random_nonce_str();

Json::FastWriter w;
const std::string &rc = w.write(request_content);

std::string authen_code = hmac_sha256(authen_key, rc); // 支付密钥计算签名

Json::Value request;
request["r"] = rc;
request["a"] = authen_code;
return w.write(request);

```

- 请求内容示例：

```

{
  "a": "6A664293966BECE1D72CA468F2B3CD8283113BE4CDF7BC0153C28F29C77054B8",
  "r": {
    "spps": [100, 200],
    "osmi": "sz0138zjfiwezee",
    "osi": "sz01FzZfei91",
    "di": "FzZfei91iefaefz",
    "ot": 1,
    "st": 1571817655,
    "et": 1571817755,
    "pn": 1,

```

```
"ps":2,
"tt":1,
"ns":"faeofeifazfeijfi"
}"
}
```

• 应答内容示例:

```
{
  "a":"C6E3976196205719D36909D706206132EFD CFC6B672E45FC05CFDE5933BB8F18",
  "rc":{"
    "tc":102,
    "ods":[
      {
        "bo":{"
          "spp":100,
          "tt":1,
          "otn":"110138591298768",
          "spp":100,
          "tt":1,
          "otn":"110138591298768",
          "tf":200,
          "ct":1571817657,
          "wx_cts":2
        }
      },
      {
        "bro":{"
          "spp":200,
          "tt":1,
          "otn":"110187443881321",
          "tf":300,
          "orn":"110187812300831",
          "rf":100,
          "ct":1571817660,
          "ali_rs":2
        }
      }
    ],
    "ns":"faeofeifazfeijfi"
  }
}
```

9.12 查询订单汇总

接口地址

https://pay.qcloud.com/cpay/brief_query_order_list_overview

请求参数

参数名	必填	类型	长度	说明
r	是	RequestContent	-	请求内容，详见本节 RequestContent。
a	是	string	64	使用支付密钥计算的认证码，目前只支持 hmac-sha256 计算认证码。

RequestContent 结构

参数名	必填	类型	长度 (Byte)	说明
spps	否	Int[]	单个元素4	sub_pay_platforms: 子支付平台列表, 100: 普通微信支付, 200: 普通支付宝, 300: 会员卡, 机具填写这三个值即可。
osmi	是	String	-	out_sub_mch_id: 云支付分配的商户 ID, 绑定成功后可获取此 ID。
osi	是	String	-	out_shop_id: 云支付分配的门店 ID, 绑定成功后可获取此 ID。
di	是	String	-	device_id: 云支付分配的设备 ID, 绑定后可获取此 ID。
ot	是	Int	4	order_type: 单据类型, 1: 支付订单, 2: 退款单, 3: 全部单据。
st	是	Int	8	start_time: 查询开始时间。
et	是	Int	8	end_time: 查询结束时间。
tt	否	Int	4	交易类型, 1: 刷卡支付, 2: 扫码支付, 3: 公众号支付, 4: App 支付, 5: 声波支付, 6: H5 支付, 8: 一码付支付, 9: 小程序支付。机具应填1。
ns	是	String	32	随机字符串, ASCII 字符 (0-9、a-z、A-Z)。

应答参数

参数名	必填	类型	长度(Byte)	说明
rc	是	ResponseContent	-	请求内容, 详见本节 ResponseContent。
a	是	String	32	使用支付密钥计算的认证码, 目前只支持 hmac-sha256 计算认证码。

ResponseContent 结构

参数名	必填	类型	长度 (Byte)	说明
s	是	Int	4	错误码。
d	是	String	最长128	即 description, 错误信息。
li	否	Int	4	即 log_id, 消息 ID。
is	是	Int	4	具体说明见 internal_status 错误码表。
brief_query_order_list	否	BriefQueryOrderListResponse	-	详见本节 BriefQueryOrderListResponse。

BriefQueryOrderResponse

参数名	必填	类型	长度(Byte)	说明
ovs	否	OrderStatClientInfo[]	-	overviews, 见 OrderStatClientInfo 说明。
ns	是	String	32	随机字符串, ASCII 字符 (0-9、a-z、A-Z)。

OrderStatClientInfo

参数名	必填	类型	长度(Byte)	说明
spp	否	Int	4	sub_pay_platform: 子支付平台, 100: 普通微信支付, 200: 普通支付宝, 300: 会员卡。
tt	否	Int	4	trade_type: 交易类型, 1: 刷卡支付, 2: 扫码支付, 3: 公众号支付, 4: App 支

				付，5：声波支付，6：H5 支付，8：一码付支付，9：小程序支付。
sc	是	Int	8	success_count：（交易量）不含撤单的，扣款成功交易笔数，有可能是负值。
sa	是	Int	8	success_amount：不含撤单的，扣款成功交易金额，有可能是负值。
sta	是	Int	8	settlement_amount：操作人扣款成功结算金额 - 操作人退款结算金额 - 操作人扣款成功撤单结算金额。
da	是	Int	8	discount_amount：必返回 net_amount - settle_amount。
psa	是	Int	8	pay_settle_amount，扣款成功结算金额。
rsa	是	Int	8	refund_settle_amount，扣款成功结算金额。
rcc	是	Int	8	refund_create_count，退款发起笔数。
rca	是	Int	8	refund_create_amount，退款发起总额。
pdg	是	Int	8	poundage，扣款成功手续费 - 撤单返还手续费 - 退款成功返还的手续费，有可能是负值。
ia	是	Int	8	income_amount，入账金额。
ora	是	Int	8	order_refunded_amount，订单已退金额。

示例说明

- 请求生成示例代码：

```
Json::Value request_content;
request_content["spps"].append(100);
request_content["spps"].append(200);
request_content["osmi"] = "sz0138zjfiwezee";
request_content["osi"] = "sz01FzZfei91";
request_content["di"] = "FzZfei91iefaefz";
request_content["ot"] = 1;
request_content["st"] = 1571817655;
request_content["et"] = 1571817755;
request_content["tt"] = 1;
request_content["ns"] = generate_random_nonce_str();

Json::FastWriter w;
const std::string &rc = w.write(request_content);

std::string authen_code = hmac_sha256(authen_key, rc); // 支付密钥计算签名

Json::Value request;
request["r"] = rc;
request["a"] = authen_code;
return w.write(request);
```

- 请求内容示例：

```
{
  "a": "156073D9F354BF8000E1EF82AC82EDB2836A9D6DF7CDFS33B05ED5DFE3407516",
  "r": {
    "spps": [100, 200],
    "osmi": "sz0138zjfiwezee",
    "osi": "sz01FzZfei91",
    "di": "FzZfei91iefaefz",
```

```
"ot":1,
"st":1571817655,
"et":1571817755,
"tt":1,
"ns":"faeofeifazfeijfi"
}"
}
```

• 应答内容示例:

```
{
  "a":"883E4B4406668C45EECCC331A8967A4A64F07BA79B4F24395A39192F28D0F231",
  "rc":{"
    "ovs":[
      {
        "spp":200,
        "tt":1,
        "sc":4,
        "sa":2400,
        "sta":2400,
        "da":0,
        "psa":2400,
        "rsa":2100,
        "rcc":1,
        "rca":300,
        "pdg":200,
        "ia":1900,
        "ora":300
      },
      {
        "spp":100,
        "tt":1,
        "sc":14,
        "sa":9100,
        "sta":9100,
        "da":100,
        "psa":9100,
        "rsa":8100,
        "rcc":1,
        "rca":300,
        "pdg":200,
        "ia":8000,
        "ora":500
      }
    ],
    "ns":"ZMEEOPPfeaeiwf122"
  }"
}
```

9.13 其它消息体说明

WxpayOrderContentExt

参数名	必填	类型	长度(Byte)	说明
cts	是	Int	4	订单当前状态，详见 WxpayOrderState。
tsd	是	String	256	对当前查询订单状态描述和下一步操作指引。

AlipayOrderContentExt

参数名	必填	类型	长度(Byte)	说明
cts	是	Int	4	订单当前状态，详见 AlipayOrderState。

CardOrderContentExt

参数名	必填	类型	长度(Byte)	说明
cts	是	Int	4	订单当前状态，详见 CardOrderState。

初始密钥认证信息

参数名	必填	类型	长度(Byte)	说明
a	是	Authen	-	认证信息，见本节 Authen。

Authen 结构

参数名	必填	类型	长度(Byte)	说明
authen_type	是	Int	4	签名类型1: HMAC-SHA256。
authen_code	是	String	32	使用机具初始密钥计算的认证码，初始密钥说明见4.2节。

支付密钥认证信息

参数名	必填	类型	长度(Byte)	说明
a	是	Authen	-	认证信息，见本节 Authen。

Authen 结构

参数名	必填	类型	长度(Byte)	说明
authen_type	是	Int	4	签名类型1: HMAC-SHA256。
authen_code	是	String	32	使用机具初始密钥计算的认证码，支付密钥说明见4.3节。

Status

枚举值	操作结果	返回内容是否带认证码	原请求是否能重试	用户操作建议
0	成功	是	是	-
3	未知	否	是	原请求重试。
101	失败	否	否	根据 description 内容，检查调用逻辑是否有问题，如认证码计算错误。
102	失败	是	否	换新单号重试，并根据 description 字段内容，检查调用逻辑是否有问题，如单号重复。
103	失败	是	是	隔3秒后原请求重试或查询结果。
104	失败	是	否	根据 description 字段内容操作，如退款时顾客余额不足。

WxpayOrderState

枚举值	说明
-----	----

1	订单初始态。
2	刷卡支付，成功。
3	统一下单，支付成功。
4	已转入退款。
5	刷卡支付，顾客停止支付。
6	统一下单，待顾客支付。
7	统一下单，订单已关闭。
8	刷卡支付，已撤单。
9	刷卡支付，用户支付中。
10	刷卡支付，支付错误。
11	作废状态，表示本地有，第三方支付平台没有的订单。

AlipayOrderState

枚举值	说明
1	订单初始态。
2	成功。
4	等待用户支付。
5	已关闭，或者已退款。
6	交易结束，不可退款。
7	订单不存在。

CardOrderState

枚举值	说明
1	订单初始态。
2	成功。
3	等待用户支付。
4	已退款。
5	已关单。

PayPlatform

枚举值	说明
1	微信支付。
2	支付宝。
3	会员卡。

WxpayRefundOrderState

--

枚举值	说明
1	退款单初始态。
2	退款成功。
3	退款失败。
4	退款处理中。
5	转入代发，退款到银行发现用户的卡作废或冻结了，导致原路退款银行卡失败，资金回流到子商户的现金账号，需要子商户人工干预，通过线下或者财付通转账的方式进行退款。
6	作废状态，表示本地有，第三方支付平台没有的订单。

AlipayRefundOrderState

枚举值	说明
1	退款单初始态。
2	退款成功。
3	退款失败。
4	退款处理中。
5	订单不存在。

CardRefundOrderState

枚举值	说明
1	退款单初始态。
2	退款成功。
3	退款失败。

10 加解密相关说明

10.1 AES-128-CBC 解密说明

- 算法：AES-128
- 模式：CBC
- 填充类型：PKCS5Padding
- 盐（salt）：要解密的内容前8字节为盐，剩余字节内容为原始明文内容的加密内容。

示例代码如下：

```
bool aes_128_cbc_decrypt(
    const std::string &session_key,
    const std::string &input,
    std::string *output)
{
    assert(output);
    output->clear();
    output->reserve(input.length() + 24); // Space for salt + padding.

    unsigned char salt[8];
    size_t total = input.size();
    const char *inptr = input.data();
```

```
if (input.length() < sizeof(salt)) {
    return false;
}

memcpy(salt, inptr, sizeof(salt));
inptr += sizeof(salt);
total -= sizeof(salt);

unsigned char iv[16];
unsigned char key[16];
int ret = EVP_BytesToKey(EVP_aes_128_cbc(), EVP_sha1(), salt,
                        (const unsigned char *)session_key.data(),
                        session_key.length(),
                        1, key, iv);

if (ret != sizeof(key)) {
    return false;
}

EVP_CIPHER_CTX *ctx = EVP_CIPHER_CTX_new();
EVP_CIPHER_CTX_init(ctx);

if (!EVP_CipherInit_ex(ctx, EVP_aes_128_cbc(), NULL, key, iv, 0)) {
    EVP_CIPHER_CTX_free(ctx);
    return false;
}

int outlen;
unsigned char buffer[2560];
for (size_t i = 0; i < total; i += 2048, inptr += 2048) {
    outlen = sizeof(buffer);
    unsigned char *in = (unsigned char *)inptr;
    int inlen = total - i < 2048 ? total - i : 2048;
    if (!EVP_CipherUpdate(ctx, buffer, &outlen, in, inlen)) {
        EVP_CIPHER_CTX_free(ctx);
        return false;
    }

    output->append(reinterpret_cast<const char *>(buffer), outlen);
}

outlen = sizeof(buffer);
if (!EVP_CipherFinal_ex(ctx, buffer, &outlen)) {
    EVP_CIPHER_CTX_free(ctx);
    return false;
}

output->append(reinterpret_cast<const char *>(buffer), outlen);
EVP_CIPHER_CTX_free(ctx);
return true;
}
```