

微信云支付

开发文档

产品文档



腾讯云

【版权声明】

©2013-2018 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

文档目录

开发文档

- SDK 简介

- HTTPS API 说明

- API 接入流程

开发文档

SDK 简介

最近更新时间：2018-10-15 12:09:09

为什么提供 SDK

- 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，且定位故障更快速和准确。

SDK 提供了哪些接口

- 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。

SDK 支持的第三方支付平台

- 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#
- 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得到；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。

HTTPS API 说明

最近更新时间：2018-08-02 16:28:18

说明

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）。
- 请求必须传认证或签名信息。其中退款请求，传签名和签名算法，其他请求传认证码和认证算法。
- 对响应要验证认证码。
- 所有接口参数名使用的字母均为小写。

发送请求举例（使用 libcurl 实现）

```
/*
  用于接受响应数据的回调函数
*/
size_t recv_data(char *ptr, size_t size, size_t nmemb, void *parm)
{
  size_t length = size * nmemb;
  std::string *data = (std::string*)parm;
  data->append(ptr, length);
  return length;
}

/*
  将 request 用 POST 方式发送到 url，响应包填充到 response 指向的 string 中
  返回是否 POST 请求是否成功
*/
bool post(const std::string &request, const std::string &url, std::string *response)
{
  CURL *hnd = curl_easy_init();

  curl_easy_setopt(hnd, CURLOPT_CUSTOMREQUEST, "POST");
  curl_easy_setopt(hnd, CURLOPT_URL, url);

  struct curl_slist *headers = curl_slist_append(NULL, "content-type: application/json");
  curl_easy_setopt(hnd, CURLOPT_HTTPHEADER, headers);

  curl_easy_setopt(hnd, CURLOPT_POSTFIELDS, request);

  // 设置云支付根证书
  curl_easy_setopt(hnd, CURLOPT_SSL_VERIFYPEER, 1);
  curl_easy_setopt(hnd, CURLOPT_SSL_VERIFYHOST, 2);
  curl_easy_setopt(hnd, CURLOPT_CAINFO, "./cloudpayrootca.pem");
}
```

```
curl_easy_setopt(hnd, CURLOPT_WRITEFUNCTION, recv_data);
std::string rc;
curl_easy_setopt(hnd, CURLOPT_WRITEDATA, (void *)&rc);

CURLcode ret = curl_easy_perform(hnd);
if (CURLE_OK != ret) {
    curl_slist_free_all(headers);
    curl_easy_cleanup(hnd);
    return false;
}
*response = rc;

curl_slist_free_all(headers);
curl_easy_cleanup(hnd);
return true;
}
```

数据包格式说明

- 请求包含两个字段：authen_info和request_content，前者表示签名或认证信息，后者表示请求具体内容，均为 json 结构。
- 响应包含两个字段：authen_info和response_content，前者表示认证信息，后者表示响应具体内容，均为 json 结构。
- 签名生成算法：RSASSA-PSS-2048-SHA256，私钥为服务商在云支付录入商户时，在网页上生成的签名私钥（该私钥只有服务商知道，云支付不知道，请妥善保管）。
- 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，在网页上生成的认证密钥。
- 如果不填非必填字段，则不要设置该字段，如需清空该字段，需上传内容为空的该字段。

计算认证码举例（使用 OpenSSL 实现）

```
/*
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, "%x", md[i]);
}
hmac->assign(format_md);

// 转大写
transform(hmac->begin(), hmac->end(), hmac->begin(), ::toupper);
return true;
}
```

计算签名举例 (使用 OpenSSL 实现)

```
/*
对计算得到的签名进行 base64 编码之后输出
返回是否成功, 成功时签名存放于sign_base64encode指向的string
*/
bool calc_RSASSA_PSS_2048_SHA256(const std::string &key,
const std::string &content,
std::string *sign_base64encode)
{
    unsigned char digest[SHA256_DIGEST_LENGTH] = {0}; //32 bytes
    int digest_len = sizeof(digest);

    std::shared_ptr<BIO> bio(BIO_new_mem_buf((void *)key.c_str(), (int)key.length()), BIO_free);
    if (!bio) {
        return false;
    }

    std::shared_ptr<RSA> rsa(PEM_read_bio_RSAPrivateKey(bio.get(), NULL, NULL, NULL), RSA_free);
    if (!rsa) {
        return false;
    }

    EVP_MD_CTX md_ctx; //当前使用 1.0.2e 版本
    EVP_MD_CTX_init (&md_ctx);

    if (!EVP_DigestInit(&md_ctx, EVP_sha256()) ||
        !EVP_DigestUpdate(&md_ctx, (const void*)content.c_str(), content.length()) ||
        !EVP_DigestFinal(&md_ctx, digest, (unsigned int *)&digest_len)) {

        EVP_MD_CTX_cleanup(&md_ctx);
        return false;
    }

    EVP_MD_CTX_cleanup(&md_ctx);
}
```

```
unsigned char em[256] = {0};
unsigned char sign[256] = {0};
int status = RSA_padding_add_PKCS1_PSS(rsa.get(), em, digest, EVP_sha256(), -2 /* maximum salt length*);
if (!status) {
return false;
}

status = RSA_private_encrypt(sizeof(em), em, sign, rsa.get(), RSA_NO_PADDING);
if (-1 == status) {
return false;
}

*sign_base64encode = base64_encode(sign, sizeof(sign));
return true;
}
```

构造请求举例（以刷卡支付为例）

```
{
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"] = "sz0100lmnx20171228151031";
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"] = "183.15.244.75";
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);

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;
}
```

响应举例（以刷卡支付为例）

- 把响应包从 string 转成 json，取出 json 里面的 response_content 和 authen_info，具体如下：

```
{
  "response_content":{
    "status":0,
    "description":"",
    "log_id":18654852,
    "internal_status":0,
    "micro_pay":{
      "pay_mch_key":{
        "pay_platform":2,
        "mch_id":"1900007941",
        "sub_mch_id":"1900008341",
        "out_mch_id":"1234mcWYS3iM5TjKLorAZ",
        "out_sub_mch_id":"12343ycHpBDv8GX]fmSvT7",
        "out_shop_id":"1234ruQClеTa9w30AaAH"
      },
      "order_content":{
        "out_trade_no":"12341008b320170802191960015",
        "transaction_id":"2017080221001004620281091091",
        "trade_type":1,
        "author_code":"282129340414399818",
        "time_expire":1501676124,
        "time_end":1501676005,
        "nonce_str":"542AB309ECA042FE92355BDEC4E2D733",
        "create_time":1501676004,
        "last_update_time":1501676005,
        "is_transforming":false,
```

```

"total_fee":1,
"fee_type":"CNY",
"cash_fee":1,
"settlement_total_fee":1,
"body":"生活用品套餐",
"alipay_order_content_ext":{
"current_trade_state":2,
"fund_bill_list":[
{"fund_channel": "ALIPAYACCOUNT","amount": 1}
],
"point_amount":0,
"invoice_amount":1
}
},
"nonce_str":"SmM10CXPIZLaIY9PIYdVGVgxc58wDRG"
}
}],
"authen_info":{
"a":{
"authen_type":1,
"authen_code":"ACD4C1920A6C8646B395D0CBB4AF9B395AC0601D1883D8EF2D7BD7238C2991A5"
}
}
}
    
```

- 对 response_content 计算验证码，并将该验证码与 authen_info 的 authen_code 进行比较。

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了“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。

交易接口

刷卡支付

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----|----|----|----|
| | | | |

| | | | |
|-----------------|---|----------------|---------------------------------|
| request_content | 是 | RequestContent | 请求内容，详见 本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

RequestContent 结构

| 参数名 | 必填 | 类型 | 说明 |
|--------------|----|-------------|---------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息，详见PayMchKey |
| pay_content | 是 | PayContent | 订单信息，详见PayContent |
| order_client | 是 | OrderClient | 客户端信息，详见OrderClient |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| 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AuthenInfo |

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) | 错误码为 407 时，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。其余错误码不用关注。 |
| 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 |
|-------------|----|------------|------------------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| pay_mch_key | 否 | PayMchKey | 支付商户信息，status 为 0 时必填。详见 PayMchKey |

| | | | |
|---------------|---|--------------|------------------------------------|
| order_content | 否 | OrderContent | 订单信息，status 为 0 时必填。详见OrderContent |
|---------------|---|--------------|------------------------------------|

构造刷卡支付请求例子

```
/*
构造请求字符串
*/
std::string gen_cloud_pay_micropay(
const int &pay_platform,
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &out_trade_no,
const std::string &author_code,
const int64_t &total_fee,
const std::string &fee_type,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,
const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const std::string &authen_key
)
{
Json::Value request_content;
request_content["nonce_str"] = generate_random_nonce_str();

Json::Value pay_mch_key, pay_content, order_client;

pay_mch_key["pay_platform"] = pay_platform;
pay_mch_key["out_mch_id"] = out_mch_id;
pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
pay_mch_key["out_shop_id"] = out_shop_id;
request_content["pay_mch_key"] = pay_mch_key;

pay_content["out_trade_no"] = out_trade_no;
pay_content["author_code"] = author_code;
pay_content["total_fee"] = total_fee;
pay_content["fee_type"] = fee_type;
request_content["pay_content"] = pay_content;

order_client["device_id"] = device_id;
order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
order_client["spbill_create_ip"] = spbill_create_ip;
```

```

request_content["order_client"] = order_client;

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

Json::Value authen_info, a;
a["authen_type"] = 1;
// 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
std::string authen_code;
if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
// 计算失败
return "";
}
a["authen_code"] = authen_code;
authen_info["a"] = a;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL
使用了发送请求举例（使用libcurl实现）中的post函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/micro_pay", &response);
    
```

扫码支付

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|---------------------------------|
| request_content | 是 | RequestContent | 请求内容，详见 本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见 AuthenInfo |

RequestContent结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------|----|-----------|---------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息，详见 PayMchKey |

| | | | |
|--------------|---|-------------|----------------------|
| pay_content | 是 | PayContent | 订单信息，详见 PayContent |
| order_client | 是 | OrderClient | 客户端信息，详见 OrderClient |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| 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AuthenInfo |

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 |
| scan_code_pay | 否 | ScanCodePayResponse | authen_info 存在时必填。详见 ScanCodePayResponse |

ScanCodePayResponse 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------|----|------------|-------------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| code_url | 否 | String(64) | status为 0 时必填。用于扫码支付时转换成支付二维码 |

构造扫码支付请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_scan_code_pay(
const int &pay_platform,
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &out_trade_no,

```

```

const int64_t &total_fee,
const std::string &fee_type,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,
const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const std::string &authen_key
)
{
    Json::Value request_content;
    request_content["nonce_str"] = generate_random_nonce_str();

    Json::Value pay_mch_key, pay_content, order_client;

    pay_mch_key["pay_platform"] = pay_platform;
    pay_mch_key["out_mch_id"] = out_mch_id;
    pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
    pay_mch_key["out_shop_id"] = out_shop_id;
    request_content["pay_mch_key"] = pay_mch_key;

    pay_content["out_trade_no"] = out_trade_no;
    pay_content["total_fee"] = total_fee;
    pay_content["fee_type"] = fee_type;
    request_content["pay_content"] = pay_content;

    order_client["device_id"] = device_id;
    order_client["staff_id"] = staff_id;
    order_client["terminal_type"] = terminal_type;
    order_client["machine_no"] = machine_no;
    order_client["sdk_version"] = sdk_version;
    order_client["spbill_create_ip"] = spbill_create_ip;
    request_content["order_client"] = order_client;

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

    Json::Value authen_info, a;
    a["authen_type"] = 1;
    // 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
    std::string authen_code;
    if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
        // 计算失败
        return "";
    }
    a["authen_code"] = authen_code;
    authen_info["a"] = a;

```

```

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过 POST 方法发送到云支付接口对应的 URL
使用了发送请求举例（使用libcurl实现）中的 post 函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/scan_code_pay", &response);
    
```

申请退款

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|--------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节 RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见 AuthenInfo |

RequestContent 结构

| 参数名 | 必填 | 类型 | 说明 |
|----------------|----|---------------|-----------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息，详见 PayMchKey |
| refund_content | 是 | RefundContent | 订单信息，详见 RefundContent |
| order_client | 是 | OrderClient | 客户端信息，详见 OrderClient |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| 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 AuthenInfo |

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 |
| 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 |
|----------------------|----|--------------------|--|
| nonce_str | 是 | String(32) | 随机字符串 |
| pay_mch_key | 否 | PayMchKey | 支付商户信息, status 为 0 时必填。详见 PayMchKey |
| refund_order_content | 否 | RefundOrderContent | 订单信息, status 为 0 时必填。详见 RefundOrderContent |

构造申请退款请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_refund(
    const int &pay_platform,
    const std::string &out_mch_id,
    const std::string &out_sub_mch_id,
    const std::string &out_shop_id,
    const std::string &out_trade_no,
    const std::string &out_refund_no,
    const int64_t &total_fee,
    const int64_t &refund_fee,
    const std::string &refund_fee_type,
    const std::string &device_id,
    const std::string &staff_id,
    const int &terminal_type,
    const std::string &machine_no,
    const std::string &sdk_version,
    const std::string &spbill_create_ip,
    const std::string &signing_key
)
    
```

```
{
Json::Value request_content;
request_content["nonce_str"] = generate_random_nonce_str();

Json::Value pay_mch_key, pay_content, order_client;

pay_mch_key["pay_platform"] = pay_platform;
pay_mch_key["out_mch_id"] = out_mch_id;
pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
pay_mch_key["out_shop_id"] = out_shop_id;
request_content["pay_mch_key"] = pay_mch_key;

pay_content["out_trade_no"] = out_trade_no;
pay_content["out_refund_no"] = out_refund_no;
pay_content["total_fee"] = total_fee;
pay_content["refund_fee"] = refund_fee;
pay_content["refund_fee_type"] = refund_fee_type;
request_content["pay_content"] = pay_content;

order_client["device_id"] = device_id;
order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
order_client["spbill_create_ip"] = spbill_create_ip;
request_content["order_client"] = order_client;

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

Json::Value authen_info, s;
s["sign_type"] = 1;
// 使用计算签名举例 (使用OpenSSL实现) 中的函数计算签名
std::string signature;
if (!calc_RSASSA_PSS_2048_SHA256(signing_key, rc, &signature)) {
// 计算失败
return "";
}
s["signature"] = signature;
authen_info["s"] = s;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后, 将请求通过 POST 方法发送到云支付接口对应的 URL
```

使用了发送请求举例 (使用 libcurl 实现) 中的 post 函数

```
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/refund", &response);
```

关闭订单

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|---------------------------|
| request_content | 是 | RequestContent | 请求内容, 详见本节 RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息, 详见 AuthenInfo |

RequestContent 结构

| 参数名 | 必填 | 类型 | 说明 |
|--------------|----|-------------|--------------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息, 详见 PayMchKey |
| order_client | 是 | OrderClient | 客户端信息, 详见 OrderClient |
| 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 |
| nonce_str | 是 | String(32) | 随机字符串 |
| trade_type | 是 | TradeType | 交易类型, 枚举值详见 TradeType |

返回参数

| 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 AuthenInfo |

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 |
| 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 |

构造关闭订单请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_close_order(
    const int &pay_platform,
    const std::string &out_mch_id,
    const std::string &out_sub_mch_id,
    const std::string &out_shop_id,
    const std::string &out_trade_no,
    const int &trade_type,
    const std::string &device_id,
    const std::string &staff_id,
    const int &terminal_type,
    const std::string &machine_no,
    const std::string &sdk_version,
    const std::string &spbill_create_ip,
    const std::string &authen_key
)
{
    Json::Value request_content;
    request_content["nonce_str"] = generate_random_nonce_str();
    request_content["out_trade_no"] = out_trade_no;
    request_content["trade_type"] = trade_type;

    Json::Value pay_mch_key, order_client;

    pay_mch_key["pay_platform"] = pay_platform;
    pay_mch_key["out_mch_id"] = out_mch_id;
    pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
    pay_mch_key["out_shop_id"] = out_shop_id;
    request_content["pay_mch_key"] = pay_mch_key;

    order_client["device_id"] = device_id;
    
```

```

order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
order_client["spbill_create_ip"] = spbill_create_ip;
request_content["order_client"] = order_client;

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

Json::Value authen_info, s;
a["authen_type"] = 1;
// 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
std::string authen_code;
if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
// 计算失败
return "";
}
a["authen_code"] = authen_code;
authen_info["a"] = a;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过 POST 方法发送到云支付接口对应的URL
使用了发送请求举例（使用libcurl实现）中的post函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/reverse", &response);
    
```

查询订单

接口地址

`https://pay.qcloud.com/cpay/query_order`

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|-------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

RequestContent结构

| 参数名 | 必填 | 类型 | 说明 |
|--------------|----|-------------|-------------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息，详见PayMchKey |
| order_client | 是 | OrderClient | 客户端信息，详见OrderClient |
| 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 |
| trade_type | 是 | TradeType | 交易类型，枚举值详见TradeType |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| 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AuthenInfo |

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 | 否 | QueryOrderResponse | authen_info存在时必填。详见QueryOrderResponse |

QueryOrderResponse结构

| 参数名 | 必填 | 类型 | 说明 |
|---------------|----|--------------|---------------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| pay_mch_key | 否 | PayMchKey | 支付商户信息，status为0时必填。详见PayMchKey |
| order_content | 否 | OrderContent | 订单信息，status为0时必填。详见OrderContent |

构造查询订单请求例子

```
/*
构造请求字符串
*/
std::string gen_cloud_pay_query_order(
const int &pay_platform,
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &out_trade_no,
const int &trade_type,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,
const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const std::string &authen_key
)
{
Json::Value request_content;
request_content["nonce_str"] = generate_random_nonce_str();
request_content["out_trade_no"] = out_trade_no;
request_content["trade_type"] = trade_type;

Json::Value pay_mch_key, order_client;

pay_mch_key["pay_platform"] = pay_platform;
pay_mch_key["out_mch_id"] = out_mch_id;
pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
pay_mch_key["out_shop_id"] = out_shop_id;
request_content["pay_mch_key"] = pay_mch_key;

order_client["device_id"] = device_id;
order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
order_client["spbill_create_ip"] = spbill_create_ip;
request_content["order_client"] = order_client;

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

Json::Value authen_info, s;
a["authen_type"] = 1;
// 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
std::string authen_code;
if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
// 计算失败
```

```

return "";
}
a["authen_code"] = authen_code;
authen_info["a"] = a;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL
使用了发送请求举例（使用libcurl实现）中的post函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/query_order", &response);
    
```

查询退款单

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|-------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

RequestContent结构

| 参数名 | 必填 | 类型 | 说明 |
|---------------|----|-------------|--------------------------|
| pay_mch_key | 是 | PayMchKey | 支付商户信息，详见PayMchKey |
| order_client | 是 | OrderClient | 客户端信息，详见OrderClient |
| 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 |
| out_refund_no | 是 | String(32) | 由客户端生成的退款单号，前缀必须是云支付订单前缀 |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

| 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AuthenInfo |

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_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 |
|----------------------|----|----------------------|---------------------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| pay_mch_key | 否 | PayMchKey | 支付商户信息，status为0时必填。详见PayMchKey |
| refund_order_content | 否 | RefundOrderContent[] | 订单信息，status为0时必填。详见RefundOrderContent |

构造查询退款单请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_query_refund_order(
const int &pay_platform,
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &out_trade_no,
const std::string &out_refund_no,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,

```

```

const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const std::string &authen_key
)
{
    Json::Value request_content;
    request_content["nonce_str"] = generate_random_nonce_str();
    request_content["out_trade_no"] = out_trade_no;
    request_content["out_refund_no"] = out_refund_no;

    Json::Value pay_mch_key, order_client;

    pay_mch_key["pay_platform"] = pay_platform;
    pay_mch_key["out_mch_id"] = out_mch_id;
    pay_mch_key["out_sub_mch_id"] = out_sub_mch_id;
    pay_mch_key["out_shop_id"] = out_shop_id;
    request_content["pay_mch_key"] = pay_mch_key;

    order_client["device_id"] = device_id;
    order_client["staff_id"] = staff_id;
    order_client["terminal_type"] = terminal_type;
    order_client["machine_no"] = machine_no;
    order_client["sdk_version"] = sdk_version;
    order_client["spbill_create_ip"] = spbill_create_ip;
    request_content["order_client"] = order_client;

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

    Json::Value authen_info, s;
    a["authen_type"] = 1;
    // 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
    std::string authen_code;
    if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
        // 计算失败
        return "";
    }
    a["authen_code"] = authen_code;
    authen_info["a"] = a;

    Json::Value request;
    request["request_content"] = rc;
    request["authen_info"] = authen_info;

    return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL

```

使用了发送请求举例（使用libcurl实现）中的post函数

```
*/  
std::string response;  
post(request, "https://pay.qcloud.com/cpay/query_refund_order", &response);
```

支付成功回调

接口地址

服务商在云支付管理后台配置的回调地址（HTTPS）

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|-------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

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PayMchKey |
| order_content | 是 | OrderContent | 订单信息，详见OrderContent |
| order_client | 是 | OrderClient | 订单信息，详见OrderClient |

返回参数

| 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AuthenInfo |

ResponseContent结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------|----|-------------|----------------------------------|
| status | 是 | Number(32) | 错误码。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 |

退款成功回调

接口地址

服务商在云支付管理后台配置的回调地址 (HTTPS)

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|----------------------------------|
| request_content | 是 | RequestContent | 请求内容, 详见 本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息, 详见AuthenInfo |

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PayMchKey |
| refund_order_content | 是 | RefundOrderContent[] | 订单信息。详见RefundOrderContent |

返回参数

| 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AuthenInfo |

ResponseContent 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------|----|-------------|---|
| status | 是 | Number(32) | 错误码。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 |

多条件查询订单信息

接口地址

- https://pay.qcloud.com/cpay/client_order_detail
- content_type : application/json

输入参数

| 参数名 | 是否必填 | 类型 | 说明 |
|-----|------|----|----|
|-----|------|----|----|

| 参数名 | 是否必填 | 类型 | 说明 |
|-----------------|------|----------------|-------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

RequestContent结构

| 参数名 | 是否必填 | 类型 | 说明 |
|------------------|------|------------|--------------------------------------|
| pay_platform | 否 | Number(32) | 第三方支付平台，详见PayPlatform |
| out_sub_mch_id | 是 | String | 云支付分配的子商户ID |
| out_shop_id | 否 | String | 云支付分配的门店全局ID |
| staff_id | 否 | String | 门店内店员的编号 |
| device_id | 否 | String | 子商户自定义的终端设备编号 |
| query_order_type | 是 | Number(32) | 查询订单类型; 详细定义见本节QueryOrderType ; 默认为3 |
| start_time | 否 | Number(64) | 查询开始时间; unix时间戳; 默认为0 |
| end_time | 是 | Number(64) | 查询结束时间; unix时间戳; 默认为当前时间 |
| page_num | 是 | Number(32) | 页码 (从1开始计数) |
| page_size | 是 | Number(32) | 单页条数 |
| nonce_str | 否 | String(32) | 随机字符串 |

QueryOrderType取值：

- 1：订单
- 2：退款单
- 3：订单和退款单

注意：

- 如果需要查询子商户的订单，out_shop_id, staff_id, device_id 不传
- 如果需要查询门店的订单，out_shop_id 必传；staff_id, device_id 不传
- 如果需要查询店员的订单，out_shop_id, staff_id 都必传

返回参数

| 参数名 | 是否必填 | 类型 | 说明 |
|------------------|------|-----------------|--------------------------|
| response_content | 是 | ResponseContent | 请求内容，详见本节ResponseContent |

| 参数名 | 是否必填 | 类型 | 说明 |
|-------------|------|------------|-------------------|
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

ResponseContent结构

| 参数名 | 是否必填 | 类型 | 说明 |
|--------------------|------|--------------------------|----------------|
| status | 是 | Status | 错误码，详见Status |
| description | 否 | String(255) | 错误描述信息 |
| 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 |
| order_detail_query | 否 | OrderDetailQueryResponse | 订单信息，详细见本节 |

OrderDetailQueryResponse结构如下：

| 参数名 | 是否必填 | 类型 | 说明 |
|---------------|------|----------------|---------------------------|
| total_count | 否 | Number(32) | 符合条件的订单总条数；注意，不是本次返回的订单条数 |
| order_details | 否 | OrderDetail [] | 订单详细信息，详细见本节 |

OrderDetail结构如下：

| 参数名 | 是否必填 | 类型 | 说明 |
|-----------------|------|-----------|-------------------|
| shop_info | 否 | ShopInfo | 门店信息，详细见ShopInfo |
| shop_staff_info | 否 | StaffInfo | 店员信息，详细见StaffInfo |
| receipt | 否 | Receipt | 订单信息，详细结构如下 |

Receipt结构如下：

| 参数名 | 是否必填 | 类型 | 说明 |
|--------------|------|-------------|--------|
| order | 否 | Order | 支付订单信息 |
| refund_order | 否 | RefundOrder | 退款单信息 |

Order结构如下：

| 参数名 | 是否必填 | 类型 | 说明 |
|-----|------|----|----|
|-----|------|----|----|

| 参数名 | 是否必填 | 类型 | 说明 |
|---------------|------|--------------|----------------------|
| order_mch | 是 | OrderMch | 支付商户信息，详细见OrderMch |
| order_content | 是 | OrderContent | 订单信息，详细见OrderContent |
| order_client | 是 | OrderClient | 客户端信息，详细见OrderClient |
| authen_info | 是 | AuthenInfo | 认证信息，详细见AuthenInfo |

RefundOrder结构如下：

| 参数名 | 是否必填 | 类型 | 说明 |
|----------------------|------|--------------------|----------------------|
| refund_order_mch | 是 | OrderMch | 支付商户信息，详细见OrderMch |
| refund_order_content | 是 | RefundOrderContent | 订单信息，详细见OrderContent |
| order_client | 是 | OrderClient | 客户端信息，详细见OrderClient |
| authen_info | 是 | AuthenInfo | 认证信息，详细见AuthenInfo |

门店接口

查询门店信息

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|---------------------------------|
| request_content | 是 | RequestContent | 请求内容，详见 本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

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帐号 |
| out_sub_mch_id | 是 | String(32) | 云支付分配给子商户的帐号 |
| nonce_str | 是 | String(32) | 随即字符串 |

| | | | |
|-----------|---|------------|-----------|
| page_num | 是 | Number(32) | 页码 (从1开始) |
| page_size | 是 | Number(32) | 单页条数 |

返回参数

| 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AuthenInfo |

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_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 |
| status为0时返回以下参数 : | | | |
| shop_infos | 否 | ShopInfo [] | 子商户信息, 详见 ShopInfo |
| total_count | 是 | Number(32) | 数据总数 |

构造查询门店信息请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_query_sub_mch_shop_info(
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const int page_num,
const int page_size,

```



```

const std::string &authen_key
)
{
    Json::Value request_content;
    request_content["nonce_str"] = generate_random_nonce_str();
    request_content["out_mch_id"] = out_mch_id;
    request_content["out_sub_mch_id"] = out_sub_mch_id;
    request_content["page_num"] = page_num;
    request_content["page_size"] = page_size;

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

    Json::Value authen_info, s;
    a["authen_type"] = 1;
    // 使用计算验证码举例（使用OpenSSL实现）中的函数计算验证码
    std::string authen_code;
    if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
        // 计算失败
        return "";
    }
    a["authen_code"] = authen_code;
    authen_info["a"] = a;

    Json::Value request;
    request["request_content"] = rc;
    request["authen_info"] = authen_info;

    return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL
使用了发送请求举例（使用libcurl实现）中的post函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/query_sub_mch_shop_info", &response);
    
```

查询子商户信息

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|-------------------------|
| request_content | 是 | RequestContent | 请求内容，详见本节RequestContent |

| | | | |
|-------------|---|------------|-------------------|
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |
|-------------|---|------------|-------------------|

RequestContent结构

| 参数名 | 必填 | 类型 | 说明 |
|----------------|----|------------|---------------------------|
| out_sub_mch_id | 是 | String(32) | 云支付分配给子商户的帐号 |
| nonce_str | 是 | String(32) | 随即字符串 |
| page_num | 是 | Number(32) | 页码, 请填1, 当前只支持查询一个子商户信息 |
| page_size | 是 | Number(32) | 单页条数, 请填1, 当前只支持查询一个子商户信息 |

返回参数

| 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AuthenInfo |

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_sub_mch_info | 否 | QuerySubMchInfoResponse | authen_info存在时必填。 详见本节QuerySubMchInfoResponse |

QuerySubMchInfoResponse 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------------------|----|------------|--------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| status为0时返回以下参数： | | | |
| sub_mch_infos | 否 | SubMch[] | 子商户信息， 详见本节SubMch |
| total_count | 是 | Number(32) | 数据总数 |

SubMch结构

| 参数名 | 必填 | 类型 | 说明 |
|----------------------|----|--------------|-----------------------|
| mch_id | 是 | String(32) | 第三方支付平台分配给服务商的帐号 |
| pay_platform | 是 | PayPlatform | 第三方支付类型，详见PayPlatform |
| company_name | 是 | String(255) | 服务商在第三方平台登记的公司名称 |
| mch_sub_uin | 是 | String(32) | 子服务商uin |
| mch_sub_company_name | 是 | String(255) | 子服务商公司名 |
| sub_mch_infos | 否 | SubMchInfo[] | 子商户信息，详见SubMchInfo |
| out_mch_id | 是 | String(32) | 服务商out id |

监控上报接口

上报客户端接口监控信息

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|---------------------------------|
| request_content | 是 | RequestContent | 请求内容，详见 本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息，详见AuthenInfo |

RequestContent 结构

| 参数名 | 必填 | 类型 | 说明 |
|----------------|----|-------------|---------------------|
| out_mch_id | 是 | String(32) | 云支付给服务商的帐号 |
| out_sub_mch_id | 是 | String(32) | 云支付分配给子商户的帐号 |
| out_shop_id | 是 | String(32) | 云支付分配给门店的帐号 |
| order_client | 是 | OrderClient | 客户端信息，详见OrderClient |
| interval | 是 | Number(32) | 上报间隔，单位：s |
| | | | |

| | | | |
|---------------------------|-----|-------------------------|---|
| is_compress | 是 | bool | 客户端根据日志大小判断是否需要压缩 |
| compress_type | 否 | CompressType | 压缩算法类型，is_compress为true时必填，详见CompressType |
| compressed_monitor_info | 二选一 | String | 压缩数据 |
| uncompressed_monitor_info | | UncompressedMonitorInfo | 未压缩数据，详见UncompressedMonitorInfo |
| nonce_str | 是 | String(32) | 随机字符串 |

UncompressedMonitorInfo 结构

| 参数名 | 必填 | 类型 | 说明 |
|--------------------|----|-----------------|-----------------------------|
| client_int_results | 否 | ClientIntResult | 客户端接口调用结果，详见ClientIntResult |
| machine_info | 是 | String | cpu使用率，内存使用率，磁盘使用情况等，json结构 |

ClientIntResult 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------|----|--------------|---|
| interface | 是 | Number(32) | 接口类型，详见Interface |
| 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 | 是 | String(255) | 错误描述 |
| time_cost | 是 | Number(32) | 接口成功平均时延，单位：ms |
| start_time | 是 | Number(64) | 接口开始时间，时间戳（秒） |
| logs | 否 | String | 接口执行过程中产生的日志列表 |
| log_id | 是 | Number(32) | 后台生成的log_id，方便对齐日志 |
| domain_name | 是 | String(1024) | 服务端域名 |

返回参数

| 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AuthenInfo |

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 |
| upload_client_monitor_info | 否 | UploadClientMonitorInfoResponse | authen_info存在时，必填。详见UploadClientMonitorInfoResponse |

UploadClientMonitorInfoResponse 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------|----|------------|---------------------------------------|
| nonce_str | 是 | String(32) | 随机字符串 |
| interval | 否 | Number(32) | status为0时必填。期望上报间隔，单位：s。0表示不用改变当前上报间隔 |

构造上报客户端接口监控信息请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_upload_client_monitor_info(
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,
const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const int interval,
const std::string &machine_info,
const std::string &authen_key
)
{
Json::Value request_content;
request_content["out_mch_id"] = out_mch_id;
request_content["out_sub_mch_id"] = out_sub_mch_id;
    
```

```
request_content["out_shop_id"] = out_shop_id;

request_content["nonce_str"] = generate_random_nonce_str();
request_content["interval"] = interval;
request_content["is_compress"] = false;

Json::Value order_client;
order_client["device_id"] = device_id;
order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
order_client["spbill_create_ip"] = spbill_create_ip;
request_content["order_client"] = order_client;

Json::Value uncompressed_monitor_info, client_int_results, client_int_result1;
client_int_result1["interface"] = "scan_code_pay";
client_int_result1["status"] = 0;
client_int_result1["description"] = "description";
client_int_result1["time_cost"] = 100;
client_int_result1["start_time"] = 1505805297;
client_int_result1["log_id"] = 73648593;
client_int_result1["domain_name"] = "https://pay.qcloud.com/cpay";
client_int_results.append(client_int_result1);
uncompressed_monitor_info["machine_info"] = machine_info;
request_content["uncompressed_monitor_info"] = uncompressed_monitor_info;

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

Json::Value authen_info, s;
a["authen_type"] = 1;
// 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
std::string authen_code;
if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
// 计算失败
return "";
}
a["authen_code"] = authen_code;
authen_info["a"] = a;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL
```

使用了发送请求举例 (使用libcurl实现) 中的post函数

```
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/upload_client_monitor_info", &response);
```

上报客户端机器配置信息

接口地址

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

content_type : application/json

输入参数

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|----------------|--------------------------|
| request_content | 是 | RequestContent | 请求内容, 详见本节RequestContent |
| authen_info | 是 | AuthenInfo | 认证信息, 详见AuthenInfo |

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帐号 |
| out_sub_mch_id | 是 | String(32) | 云支付分配给子商户的帐号 |
| out_shop_id | 是 | String(32) | 云支付唯一标识门店的帐号 |
| order_client | 是 | OrderClient | 客户端信息, 详见OrderClient |
| machine_info | 是 | String | 主机信息, 如主机名, 磁盘, CPU, 内存信息等, json结构 |
| nonce_str | 是 | String(32) | 随机字符串 |

返回参数

| 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AuthenInfo |

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 |
| upload_client_conf_info | 否 | UploadClientConfInfoResponse | authen_info存在时必填。详见 UploadClientConfInfoResponse |

UploadClientConfInfoResponse 结构

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

构造上报客户端机器配置信息请求例子

```

/*
构造请求字符串
*/
std::string gen_cloud_pay_upload_client_conf_info(
const std::string &out_mch_id,
const std::string &out_sub_mch_id,
const std::string &out_shop_id,
const std::string &device_id,
const std::string &staff_id,
const int &terminal_type,
const std::string &machine_no,
const std::string &sdk_version,
const std::string &spbill_create_ip,
const std::string &machine_info,
const std::string &authen_key
)
{
Json::Value request_content;
request_content["out_mch_id"] = out_mch_id;
request_content["out_sub_mch_id"] = out_sub_mch_id;
request_content["out_shop_id"] = out_shop_id;

request_content["nonce_str"] = generate_random_nonce_str();

Json::Value order_client;
order_client["device_id"] = device_id;
order_client["staff_id"] = staff_id;
order_client["terminal_type"] = terminal_type;
order_client["machine_no"] = machine_no;
order_client["sdk_version"] = sdk_version;
    
```



```

order_client["spbill_create_ip"] = spbill_create_ip;
request_content["order_client"] = order_client;

request_content["machine_info"] = machine_info;

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

Json::Value authen_info, s;
a["authen_type"] = 1;
// 使用计算认证码举例（使用OpenSSL实现）中的函数计算认证码
std::string authen_code;
if (!calc_HMAC_SHA256(authen_key, rc, &authen_code)) {
// 计算失败
return "";
}
a["authen_code"] = authen_code;
authen_info["a"] = a;

Json::Value request;
request["request_content"] = rc;
request["authen_info"] = authen_info;

return w.write(request);
}
/*
构造请求完毕之后，将请求通过POST方法发送到云支付接口对应的URL
使用了发送请求举例（使用libcurl实现）中的post函数
*/
std::string response;
post(request, "https://pay.qcloud.com/cpay/upload_client_conf_info", &response);
    
```

公共数据结构

认证签名信息

AuthenInfo结构

| 参数名 | 必填 | 类型 | 说明 |
|-----|--------------------|-----------|-------------------|
| a | 退款接口使用签名,其他接口使用认证码 | Authen | 认证信息, 详见Authen |
| s | | Signature | 签名信息, 详见Signature |

Authen 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----|----|----|----|
|-----|----|----|----|

| | | | |
|-------------|---|--------------|--------------------|
| authen_type | 是 | Number(32) | 认证算法, 详见AuthenType |
| authen_code | 是 | String(2048) | 验证码 |

Signature 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------|----|--------------|-------------------|
| sign_type | 是 | Number(32) | 签名算法, 详见Signature |
| sign | 是 | String(2048) | 签名内容 |

订单信息

OrderContent 结构 (仅作为返回参数)

| 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, 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) |
| notify_url | 否 | String(1024) | 第三方支付平台回调url (刷卡支付不需要该字段) |
| 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) |
| is_transforming | 是 | Bool | 系统是否正在处理中 |
| 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CNY) |

| | | | |
|--------------------------|---|-----------------------|---------------------------------|
| settlement_total_fee | 否 | Number(32) | 应结支付金额，单位分 |
| body | 否 | String(128) | 商品或订单简要描述 |
| detail | 否 | String(6000) | 商品详细列表，详见Detail |
| wxpay_order_content_ext | 否 | WxpayOrderContentExt | 微信支付扩展信息，详见WxpayOrderContentExt |
| alipay_order_content_ext | 否 | AlipayOrderContentExt | 支付宝扩展信息，详见AlipayOrderContentExt |

WxpayOrderContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|---------------------|----|-----------------|---|
| 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 |
| coupon_fee | 否 | Number(32) | 代金券金额，等于支付总金额减现金支付金额 |
| coupon_count | 否 | Number(32) | 代金券数量 |
| coupon_infos | 否 | WxpayCouponInfo | 代金券信息，详见WxpayCouponInfo |
| product_id | 否 | String(32) | 商品id，子商户自定义，扫码支付时必传 |
| 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 |
| 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 |

WxpayCouponInfo 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------|----|------------|--|
| coupon_batch_id | 否 | String(20) | 代金券或立减优惠批次id |
| coupon_id | 否 | String(20) | 代金券或立减优惠id |
| 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 订单使用代金券时有返回 |

AlipayOrderContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------|----|---------------------|---|
| 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: "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 }] |
| fund_bill_list | 是 | AlipayFundBill | 支付渠道, 支付宝回包的内容, 详见AlipayFundBill, 示例: "fund_bill_list": [{ "fund_channel": "ALIPAYACCOUNT", "amount": 10, "real_amount": 11.21 }] |
| 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：[{ "goods_id":"STANDARD1026181538", "goods_name":"雪碧", "discount_amount":"100.00", "voucher_id":"2015102600073002039000002D5O" }] |
| 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 |

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 |

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 |

退款单信息

RefundOrderContent 结构 (仅作为返回参数)

| 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 | 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) |
| is_transforming | 是 | Bool | 系统是否正在处理中 |
| 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 |
| wxpay_refund_order_content_ext | 是 | WxpayRefundOrderContentExt | 微信支付扩展信息, 详见WxpayRefundOrderContentExt |
| alipay_refund_order_content_ext | 是 | AlipayRefundOrderContentExt | 支付宝扩展信息, 详见AlipayRefundOrderContentExt |

WxpayRefundOrderContentExt结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------|----|------------|---|
| 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 |
| settlement_refund_fee | 否 | Number(32) | 去掉非充值代金券退款金额后的退款金额, 单位: 分。 退款金额=申请退款金额-非充值代金券退款金额, 退款金额 <=申请退款金额 退款金额=申请退款金额-非充值代金券退款金额 退款金额 <=申请退款金额 |
| 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 |
| coupon_refund_infos | 否 | WxpayCouponInfo | 退款代金券信息，详见WxpayCouponInfo |
| 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_rcv_account | 否 | String(64) | 取当前退款单的退款入账方 1、退回银行卡：{银行名称}{卡类型}{卡尾号} 2、退回支付用户零钱:支付用户零钱 |

AlipayRefundOrderContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------------------|----|------------------------------|--------------------------------|
| 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 |
| refund_status | 是 | AlipayRefundOrderState(枚举类型) | 退款状态，详见 AlipayRefundOrderState |

查询订单时商户信息

OrderMch结构

| 参数名 | 是否必填 | 类型 | 说明 |
|------------------|------|------------|------------------------|
| pay_platform | 是 | Number(32) | 第三方支付平台；详细见PayPlatform |
| out_mch_id | 否 | String | 云支付分配的服务商ID |
| out_sub_mch_id | 否 | String | 云支付分配的子商户ID |
| out_shop_id | 否 | String | 云支付分配的门店全局ID |
| out_channel_id | 否 | String | 云支付分配给渠道商的ID |
| out_card_id | 否 | String | 会员卡ID |
| sub_mch_pay_info | 否 | String | 商户下单时存在订单中的特定信息 |

| 参数名 | 是否必填 | 类型 | 说明 |
|------------------------|------|---------------------|--------------|
| mch_uin | 否 | String | 服务商的腾讯云账号ID |
| mch_sub_uin | 否 | String | 子服务商的腾讯云账号ID |
| using_stream_sub_mch | 否 | bool | 是否使用银行商户 |
| upstream_order_mch_ext | 否 | UpstreamOrderMchExt | 银行渠道相关信息 |
| wxpay_order_mch_ext | 否 | WxpayOrderMchExt | 微信支付服务商扩展信息 |
| alipay_order_mch_ext | 否 | AlipayOrderMchExt | 支付宝服务商扩展信息 |
| card_order_mch_ext | 否 | CardOrderMchExt | 会员卡服务商扩展信息 |

WxpayOrderMchExt结构

| 参数名 | 是否必填 | 类型 | 说明 |
|------------------|------|--------|------------------|
| mch_id | 否 | String | 微信支付分配给服务商的账号 |
| sub_mch_id | 否 | String | 微信支付分配给子商户的账号 |
| shop_id | 否 | String | 微信支付分配给门店的编号 |
| app_id | 否 | String | 微信支付分配给服务商的公众号ID |
| sub_app_id | 否 | String | 微信支付分配给子商户的公众号ID |
| open_id | 否 | String | 顾客在服务商公众号下的唯一标识 |
| sub_open_id | 否 | String | 顾客在子商户公众号下的唯一标识 |
| is_subscribe | 否 | bool | 用户是否关注了服务商公众号 |
| sub_is_subscribe | 否 | bool | 用户是否关注了子商户的公众号 |
| is_bill | 否 | bool | 是否为微信买单商户 |
| use_bill | 否 | bool | 是否走微信买单渠道 |
| bill_channel_id | 否 | String | 微信买单渠道号 |
| bill_shop_id | 否 | String | 微信买单默认门店号 |
| is_macro | 否 | bool | 是否是小微商户 |

AlipayOrderMchExt结构

| 参数名 | 是否必填 | 类型 | 说明 |
|-----|------|----|----|
|-----|------|----|----|

| 参数名 | 是否必填 | 类型 | 说明 |
|------------|------|--------|-------------------|
| app_id | 否 | String | 支付宝分配给服务商的APP ID |
| sub_app_id | 否 | String | 支付宝分配给子商户的APP ID |
| user_id | 否 | String | 顾客的用户号 |
| sub_mch_id | 否 | String | 支付宝的子商户号, 银行服务商使用 |

交易请求时的商户信息

PayMchKey 结构

| 参数名 | 必填 | 类型 | 说明 |
|------------------------|----|--------------------|--------------------------------------|
| pay_platform | 是 | PayPlatform | 第三方支付类型, 详见PayPlatform |
| out_mch_id | 是 | String(32) | 云支付分配给服务商的帐号 |
| out_sub_mch_id | 是 | String(32) | 云支付分配给子商户的帐号 |
| out_shop_id | 是 | String(32) | 云支付唯一标识门店的账号 |
| notify_open_ids | 否 | String(255)[] | 关注本次操作的店员/店长在服务商微信公众号下的open_id。可选。数组 |
| wxpay_pay_mch_key_ext | 否 | WxpayPayMchKeyExt | 微信支付扩展信息 |
| alipay_pay_mch_key_ext | 否 | AlipayPayMchKeyExt | 支付宝扩展信息 |

WxpayPayMchKeyExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|---------|----|-------------|-------------------|
| open_id | 否 | String(255) | 用户在服务商微信公众号下的唯一标识 |

AlipayPayMchKeyExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|---------|----|-------------|--------------|
| user_id | 否 | String(255) | 用户在支付宝下的唯一标识 |

交易请求时的支付订单信息

PayContent 结构 (仅交易请求使用)

| 参数名 | 必填 | 类型 | 说明 |
|-----|----|----|----|
|-----|----|----|----|

| | | | |
|------------------------|---|---------------------|---|
| 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 | 否 | String(128) | 刷卡支付时的授权码（刷卡支付必填，其他不填） |
| 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） |
| total_fee | 是 | Number(32) | 订单总金额，单位：分 |
| fee_type | 是 | String(3) | 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： （1）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； （2）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； （3）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； （4）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） （5）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(6000) | 商品详细列表，由json转化而来，详见Detail。 |
| wxpay_pay_content_ext | 否 | WxpayPayContentExt | 微信支付扩展信息，详见WxpayPayContentExt |
| alipay_pay_content_ext | 否 | AlipayPayContentExt | 支付宝扩展信息，详见AlipayPayContentExt |

Detail 结构

| 参数名 | 必填 | 类型 | 说明 |
|--------------|----|---------------|---|
| cost_price | 可选 | 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 | 可选 | String(32) | 商家小票ID |
| goods_detail | 必填 | GoodsDetail[] | 商品详情，详见GoodsDetail |

GoodsDetail 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----|----|----|----|
| | | | |

| | | | |
|------------|---|-------------|--|
| goods_id | 否 | String(32) | 商品的编号 |
| goods_name | 否 | String(256) | 商品名称 |
| quantity | 是 | Number(32) | 商品数量 |
| price | 是 | Number(32) | 商品单价，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，否则会无法享受优惠 |

WxpayPayContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|------------|----|-------------|-----------------------------------|
| 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（扫码支付必填，刷卡支付不填，其他可选） |
| 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 |

AlipayPayContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------|----|-------------|--------------|
| 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 |
| product_code | 否 | String(128) | 产品码 |
| royalty_info | 否 | String(256) | 分账信息，json结构 |
| extend_params | 否 | String(256) | 扩展信息，花呗相关的逻辑 |
| disable_pay_channels | 否 | String(128) | 不可用渠道 |

退款请求时的退款单信息

RefundContent 结构（仅退款请求使用）

| 参数名 | 必填 | 类型 | 说明 |
|---------------|----|------------|--------------------------|
| 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 |
| out_refund_no | 是 | String(32) | 由客户端生成的退款单号，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 |

| | | | |
|---------------------------------|---|-----------------------------|---------------------------------------|
| refund_fee | 是 | Number(32) | 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，请填写CNY） |
| wxpay_refund_content_ext | 否 | WxpayRefundOrderContentExt | 微信支付扩展信息，详见WxpayRefundOrderContentExt |
| alipay_refund_order_content_ext | 否 | AlipayRefundOrderContentExt | 支付宝扩展信息，详见AlipayRefundOrderContentExt |

WxpayRefundOrderContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------|----|-----------------|--|
| 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 |
| settlement_refund_fee | 否 | Number(32) | 去掉非充值代金券退款金额后的退款金额，单位：分。 退款金额=申请退款金额-非充值代金券退款金额，退款金额<=申请退款金额 退款金额=申请退款金额-非充值代金券退款金额 退款金额<=申请退款金额 |
| 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 |
| coupon_refund_infos | 否 | WxpayCouponInfo | 退款代金券信息，详见WxpayCouponInfo |
| 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_rcv_account | 否 | String(64) | 取当前退款单的退款入账方 1、退回银行卡：{银行名称}{卡类型}{卡尾号} 2、退回支付用户零钱:支付用户零钱 |

AlipayRefundOrderContentExt 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------|----|-----------|----------------|
| 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 |
| refund_status | 是 | AlipayRefundOrderState(枚举类型) | 退款状态, 详见 AlipayRefundOrderState |

客户端信息

OrderClient 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------------|----|------------|--|
| device_id | 是 | String(64) | 子商户自定义, 终端设备号 |
| staff_id | 是 | 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 : iso, 4 : linux, 100 : 其他 |
| sub_terminal_type | 否 | Number(32) | 子终端类型, 代表一个机具品牌, 具体值可以联系云支付分配 |
| machine_no | 否 | String(32) | 刷卡支付特有, 每个收银终端的唯一码 (刷卡支付必填) |
| 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 |

门店信息

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 |
| device_type | 是 | Number(32) | 设备类型 1: 刷卡支付；2: 扫码支付；3: 混合支付，支持刷卡支付+扫码支付；4: 固定二维码支付 |
| remark | 否 | String(64) | 子商户自定义备注信息 |
| device_name | 是 | String | 设备名称 |

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 |

子商户信息

SubMchInfo 结构

| 参数名 | 必 | 类型 | 说明 |
|-----|---|----|----|
|-----|---|----|----|

| | 填 | | |
|------------------------------|---|--------------------|---|
| sub_mch_id | 是 | String(32) | 微信支付分配的子商户号 |
| company_name | 是 | String(255) | 子商户在第三方支付平台登记的公司名称 |
| desc | 否 | String(255) | 子商户描述 |
| cloud_cashier_id | 是 | String(32) | 云支付分配的唯一订单前缀， 下单时商户的订单号需要以这个id开头 |
| out_sub_mch_id | 是 | String(32) | 云支付分配的商户id |
| default_order_body | 否 | String | 默认的商品或订单简要描述,固定二维码支付时使用 |
| sub_mch_admin_infos | 否 | SubMchAdminInfo | 商户管理者的信息列表， 详见本节 SubMchAdminInfo |
| out_sub_mch_id_url | 是 | String(128) | 子商户二维码 |
| merchant_name | 是 | String(256) | 子商户在第三方支付平台的商户名 |
| logo | 否 | String | 商户logo |
| admin_email | 否 | String(255) | 商户管理员邮箱 |
| phone | 否 | String(255) | 商户联系电话 |
| one_code_pay_ad_info | 否 | OneCodePayAdInfo | 一码支付中，顾客完成支付后的广告信息, 详见本节OneCodePayAdInfo |
| is_use_cpay_shop_system | 否 | Bool | 是否使用云支付门店管理系统 |
| ad_page_url | 否 | String | 支付成功后广告页面url |
| buslic_id | 否 | String(32) | 营业执照 ID |
| sub_mch_source | 否 | UInt | 子商户来源 |
| direct | 是 | Bool | false 时使用 upstream_out_channel_id 做转发 |
| upstream_out_channel_id | 否 | String | 将这一商户的支付请求通过这一渠道转发至其他（相关联的）商户 |
| upstream_company_name | 否 | String | 上游服务商的公司名 |
| selectable_upstream_sub_mchs | 否 | UpstreamSubMchInfo | 可选择的上游子商户,不包含各种 key, 详见本节 UpstreamSubMchInfo |
| bank_rate | 否 | UInt | 商户进件后，银行收取的费率 1/1000000 |
| wxpay_sub_mch_info_ext | 否 | WxpaySubMchInfoExt | 微信支付子商户扩展信息, 详见本节 |

| | | | |
|-------------------------|---|---------------------|---|
| | | | WxpaySubMchInfoExt |
| alipay_sub_mch_info_ext | 否 | AlipaySubMchInfoExt | 支付宝子商户扩展信息,详见本节 AlipaySubMchInfoExt |

SubMchAdminInfo 结构

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------------|----|--------|-----------------|
| name | 是 | String | 真实姓名 |
| receive_one_code_pay_notify | 否 | Bool | 是否接收一码支付的成功消息通知 |

OneCodePayAdInfo 结构

| 参数名 | 必填 | 类型 | 说明 |
|---------|----|------------|--------------------------|
| picture | 是 | String(64) | 广告图片内容 |
| url | 否 | String(64) | 点击广告图片后的跳转链接,如没有,则图片无法点击 |

UpstreamSubMchInfo 结构

| 参数名 | 必填 | 类型 | 说明 |
|-------------------------|----|------------|--------------|
| upstream_out_sub_mch_id | 是 | String(64) | 上游云支付子商户帐号 |
| upstream_out_mch_id | 是 | String | 上游云支付服务商帐号 |
| upstream_out_channel_id | 是 | String | 上游服务商渠道id |
| cached_buslic_id | 是 | String | 上游子商户营业执照id |
| cached_cloud_cashier_id | 是 | String | 上游子商户商户订单号前缀 |
| cached_mch_company_name | 是 | String | 上游服务商的公司名 |

WxpaySubMchInfoExt 微信支付子商户扩展信息

| 参数名 | 必填 | 类型 | 说明 |
|------------|----|------------|------------------|
| sub_app_id | 否 | String(32) | 第三方支付平台分配给子商户的帐号 |

AlipaySubMchInfoExt 支付宝子商户扩展信息

| 参数名 | 必填 | 类型 | 说明 |
|-----------------------|----|------------|--------------------|
| sub_mch_user_id | 否 | String(64) | 子商户在支付宝平台的用户id即uid |
| ali_authorization_url | 否 | String | 子商户支付宝授权二维码 |

枚举值定义

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 |

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 |

WxpayOrderState 枚举变量

| 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 |

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者冻结了，导致原路退款银行卡失败，资金回流到子商户的现金帐号，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 |

认证加密信息

AuthenType 枚举变量

| 枚举值 | 说明 |
|-----|-------------|
| 1 | HMAC-SHA256 |
| 2 | MD5 |

SignType 枚举变量

| 枚举值 | 说明 |
|-----|------------------------|
| 1 | RSASSA-PSS-2048-SHA256 |

EncryptType 枚举变量

| 枚举值 | 说明 |
|-----|-------------|
| 1 | AES-128-GCM |

接口相关信息
Status 枚举变量

| 枚举值 | 说明 |
|-----|--|
| 0 | 成功。带认证码，调用者需要验证认证码是否正确 |
| 3 | 系统内部错误，操作结果未知，可重试，不带认证码 |
| 101 | 操作失败，且不建议重试，不带认证码 |
| 102 | 操作失败，且建议换新单号重试，带认证码，调用者需要验证认证码是否正确 |
| 103 | 系统内部错误，可重试，带认证码，调用者需要验证认证码是否正确 |
| 104 | 操作失败，且不建议重试。带认证码，调用者需要验证认证码是否正确 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，如是，则说明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。 |

Interface 枚举变量

| 枚举值 | 说明 |
|-----|--------|
| 14 | 查询门店信息 |
| 100 | 刷卡支付 |

| | |
|-----|-----------|
| 101 | 扫描支付 |
| 103 | 关单 |
| 104 | 退款 |
| 106 | 查询订单 |
| 107 | 查询退款单 |
| 108 | 撤单 |
| 202 | 接口监控上报 |
| 203 | 客户端机器配置上报 |

其他信息

CompressType

| 枚举值 | 说明 |
|-----|-------|
| 1 | kZip |
| 2 | kRAR |
| 3 | kGZip |

API 接入流程

最近更新时间：2018-08-02 17:34:05

文档提供了刷卡支付的接入流程，包含商户入驻说明、刷卡支付、查询订单、退款、退款查询等流程说明，帮助接入方更好地理解整个流程。

[下载地址>>](#)