

# 广东综合支付平台开放接口

## 设计说明书 V1.0.1

广东综合支付平台开放接口 .....	1
设计说明书 V1.0.1 .....	1
1、文档说明.....	6
1.1 阅读对象 .....	6
1.2 术语 .....	6
1.3 协议规则 .....	7
1.4 注意事项 .....	7
2、安全规范.....	8
数字签名 .....	8
签名原始串 .....	8
签名算法 .....	9
RSA 签名 .....	9
MD5 签名 .....	9
签名后请求参数示例 .....	9
密钥管理 .....	10
测试参数 .....	10

注意事项.....	11
3、报文格式.....	11
提交数据.....	11
3.1、请求报文.....	11
3.2、响应报文.....	12
4、补单机制.....	12
5、错误码.....	13
5.1、公共错误码.....	13
5.2、业务错误码.....	15
6、请求参数.....	15
6.1、公共请求参数.....	15
6.2、公共响应参数.....	16
公共业务响应参数.....	16
7、API 接口.....	17
7.1、统一下单.....	17
应用场景.....	17
接口链接.....	17
请求参数.....	17
响应参数.....	18
业务状态码.....	19
请求示例.....	19
响应示例.....	20
7.2、查询订单.....	21

应用场景 .....	21
接口链接 .....	21
请求参数 .....	21
响应参数 .....	21
业务错误码 .....	23
请求示例 .....	23
响应示例 .....	23
7.3、撤销订单 .....	24
应用场景 .....	24
接口链接 .....	24
请求参数 .....	24
响应参数 .....	25
业务错误码 .....	25
请求示例 .....	25
响应示例 .....	26
7.4、关闭订单 .....	26
应用场景 .....	26
接口链接 .....	26
请求参数 .....	26
响应参数 .....	27
业务错误码 .....	27
请求示例 .....	27
响应示例 .....	28

7.5、申请退款 .....	28
应用场景 .....	28
接口地址 .....	29
请求参数 .....	29
响应参数 .....	29
业务错误码 .....	30
请求示例 .....	31
响应示例 .....	31
7.6、退款查询 .....	32
应用场景 .....	32
接口地址 .....	32
请求参数 .....	32
响应参数 .....	33
业务错误码 .....	33
请求示例 .....	34
响应示例 .....	34
7.7、退款查询扩展（根据支付单号查询） .....	35
应用场景 .....	35
分页查询 .....	35
接口地址 .....	35
请求参数 .....	35
响应参数 .....	36
业务错误码 .....	36

请求示例 .....	37
响应示例 .....	37
7.8、交易结果通知 .....	38
应用场景 .....	38
接口链接 .....	39
请求参数 .....	39
响应参数 .....	39
支付结果通知示例 .....	39
退款结果通知示例 .....	40
附录一：交易类型说明 .....	40
附录二：各渠道请求扩展参数说明 .....	40
附录三：各渠道响应扩展参数说明 .....	42

# 1、文档说明

## 1.1 阅读对象

供商户或者服务商的开发人员与业务人员参考和查询。

## 1.2 术语

### 1.2.1、刷卡支付（BSC）

刷卡支付是用户展示云闪付/微信钱包/支付宝内的“刷卡条码/二维码”给商户系统扫描后直接完成支付的模式。

主要应用线下面对面收银的场景。

### 1.2.2、扫码支付（CSB）

扫码支付是商户系统按云闪付/微信支付/支付宝协议生成支付二维码，用户再用云闪付/微信/支付宝“扫一扫”完成支付的模式。该模式适用于 PC 网站支付、实体店单品或订单支付、媒体广告支付等场景。

### 1.2.3、APP 支付（APP）

又称移动端支付，是商户通过在移动端应用 APP 中集成云闪付/微信/支付宝开放 SDK 调起支付模块完成支付的模式。

### 1.2.4、公众号支付（MP）

用户在微信中打开商户的 H5 页面，商户在 H5 页面通过调用微信支付提供的 JSAPI 接口调起微信支付模块完成支付。应用场景有：

- 1) 用户在微信公众账号内进入商家公众号，打开某个主页面，完成支付
- 2) 用户的好友在朋友圈、聊天窗口等分享商家页面连接，用户点击链接打开商家页面，完成支付
- 3) 将商户页面转换成二维码，用户扫描二维码后在微信浏览器中打开页面后完成支付

### 1.2.5、H5 支付（H5）

H5 支付主要是在手机、iPad 等移动设备中通过浏览器来唤起微信支付的支付产品。

### 1.2.6、小程序支付（APPLET）

小程序支付是专门被定义使用在小程序中的支付产品。目前在小程序中能且只能使用小程序支付的方式来唤起微信支付。

## 1.3 协议规则

商户接入统一支付平台，调用 API 遵循以下规则：

传输方式	为保证交易安全性，强制采用 HTTPS 传输
提交方式	采用 POST 方法提交
数据格式	提交和返回数据都为 JSON 格式
字符编码	统一采用 UTF-8 字符编码
签名算法	MD5、RSA2，后续会兼容 SHA1、SHA256、HMAC 等。
签名要求	请求和接收数据均需要校验签名，详细方法请参考安全规范-签名算法
判断逻辑	先判断网关状态码 code，然后判断业务状态 sub_code，最后判断交易结果

## 1.4 注意事项

- 1、所有涉及到金额的单位都是分，最小的单位是 1 分，不能有小数出现
- 2、notify\_url 是平台服务器从后台直接发起请求到商户服务器，商户处理时不能检查用户的 cookie 或 session；商户更新 DB 等发货流程需要在 notify\_url 完成后，以确保掉单时，平台补单能成功补上
- 3、notify\_url 有可能重复通知，商户需要做去重处理，避免多次发货
- 4、notify\_url 收到的通知，商户处理成功或检查订单已经处理，需要返回处理成功的标志纯字符串 success，字符串 success 不区分大小写；如果我们没有收到返回的 success，我方服务器继续向你们发送通知，在三小时后将不再通知；假设所有订单都没有返回 success，将增加我们服务器的通知负载，最坏的情况，可能就是导致正常通知商户的通知发生延迟；另外我们会催促你们完善，如果长期不改善，研发或运维技术将对贵司开通的支付接口采取控制措施。
- 5、文档中请求接口时传的参数，必填为是的参数是必须要传的（如有缺少会报错），必填为否的参数可以传也可以不传
- 6、返回参数中必填为是的参数是一定会返回的，必填为否的参数则不一定返回，因升级或配置等情况实际返回参数可能不会跟文档完全一致，必须以实际接收到的参数为准
- 7、关于用户 openid：在关注者与公众号产生消息交互后，公众号可获得关注者的 OpenID（加密后的微信号，每个用户对每个公众号的 OpenID 是唯一的。对于不同公众号，同一用户的 OpenID 不同）获取 OpenID 可参考以下地址：

[https://mp.weixin.qq.com/wiki?t=resource/res\\_main&id=mp1421140842&token=&lang=zh\\_CN](https://mp.weixin.qq.com/wiki?t=resource/res_main&id=mp1421140842&token=&lang=zh_CN)

### 8、其它注意事项

#### (1) 参数大小写问题

请留意文档中要求的字符大小写问题，如“md5 运算后，字符串的字符要转换为大写”。

#### (2) 参数格式问题

所有传入参数，均为字符串类型，请注意文档中各处的具体要求。

#### (3) 时间戳问题

请使用 Linux 时间戳，注意为字符串格式。

(4) 同一商户订单号支付问题

## 2、安全规范

### 数字签名

为了保证数据传输过程中的数据真实性和完整性，我们需要对数据进行数字签名，在接收签名数据之后进行签名校验。

数字签名有两个步骤，先按一定规则拼接要签名的原始串，再选择具体的算法和密钥计算出签名结果。

### 签名原始串

无论是请求还是应答，签名原始串按以下方式组装成待签名字符串：

- 1、除 sign 字段外，所有参数按照字段名的 ascii 码从小到大排序后使用 Querystring 的格式（即 key1=value1&key2=value2...）拼接而成，空值不传递，不参与签名组串。
- 2、签名原始串中，字段名和字段值都采用原始值，不进行 URL Encode。
- 3、平台返回的应答或通知消息可能会由于升级增加参数，请验证应答签名时注意 允许这种情况。

举例：

调用某个接口，接口有如下字段

```
{  
  "charset": "UTF-8",  
  "encrypt_type": "aes",  
  "nonce_str": "329986",  
  "biz_content": {  
    "extend": {  
      "terminal_no": "10300632",  
      "auth_code": "134711323868398975"  
    },  
    "time_expire": "20201207154516",  
    "out_trade_no": "NO20201207144516370661",  
    "time_start": "20201207144516",  
    "total_amount": "1",  
    "currency": "156",  
    "client_ip": "113.71.218.142",  
    "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",  
    "notify_url": "",  
    "body": "test",  
    "trans_type": "BSC"  
  }  
}
```



```

    },
    "mer_id": "YW0014406000000",
    "format": "json",
    "version": "1.0",
    "sign_type": "RSA2",
    "timestamp": "20201207144542"
}

```

正确的签名字段排序为：

```

biz_content={"extend":{"terminal_no":"10300632","auth_code":"134711323868398975"},"time_expire":"20201207154516","out_trade_no":"NO20201207144516370661","time_start":"20201207144516","total_amount":"1","currency":"156","client_ip":"113.71.218.142","attach":{"aaano=xxxxxxxxxxxxx,bbno=xxxxxxxxxxxxx","notify_url":"","body":"test","trans_type":"BSC"}& charset=UTF-8& encrypt_type=aes& format=json& mer_id=YW0014406000000& nonce_str=329986& sign_type=RSA2& timestamp=20201207144542& version=1.0

```

## 签名算法

目前只支持 MD5、RSA2

### RSA 签名

采用私钥签名公钥验签的方式，平台和商户各持有对方公钥。

使用各自语言对应的 SHA256WithRSA(对应 sign\_type 为 RSA2)或 SHA1WithRSA(对应 sign\_type 为 RSA)签名函数利用商户私钥对待签名字符串进行签名，并进行 Base64 编码。

### MD5 签名

将“&key=签名密钥”拼接在待签名字符串后面，然后调用 MD5 摘要算法，将计算结果转大写，计算公式：

```
sign= MD5(待签名字符串&key=商户密钥).toUpperCase()
```

### 签名后请求参数示例

```

{
    "charset": "UTF-8",
    "encrypt_type": "aes",
    "nonce_str": "329986",

```

```
"biz_content": {
  "extend": {
    "terminal_no": "10300632",
    "auth_code": "134711323868398975"
  },
  "time_expire": "20201207154516",
  "out_trade_no": "NO20201207144516370661",
  "time_start": "20201207144516",
  "total_amount": "1",
  "currency": "156",
  "client_ip": "113.71.218.142",
  "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
  "notify_url": "",
  "body": "test",
  "trans_type": "BSC"
},
"mer_id": "YW0014406000000",
"format": "json",
"sign": "LwkmZf4S827ssMpHp07gmTGnE6GZfMTY2/zDwie7MrY0AAJAMNYVwqBaVrAecRrhotHiZf6mYrwIYthx7aU4Ic+kccBTbrVokAPfxk4VHB0Jn15oYPWgFWWxVC2XHKwDAMYH4wFYCC2vuPZEAQ4973fgOp0+7Ys08X/DjqT8bj0iuKjtqUi0a50UUq5ZAD4PeooJsmL1LE3eHEXH9iDK93HHvHaPhiHzJkLkClEqjGcUAuAYIqxeaXoYE13Bd0SkjBsxQULvGY4UBwxRr/g6gMZE0B6vy06kzB6p+YjAw9qh10igqyZ3F3dEaiAK5P1nnsNrtvsEZSLojbpYX99i+Q=",
"version": "1.0",
"sign_type": "RSA2",
"timestamp": "20201207144542"
}
```

## 密钥管理

平台密钥由系统自动生成公私钥对，私钥系统保留，平台公钥由相关对接人员负责下发给商户。

商户公私钥对由商户自主生成，并提供公钥和签名类型（目前仅支持 RSA2）给到相关人员，由相关人员上传至统一支付平台。

## 测试参数

请联系相关对接人员获取。

## 注意事项

公共参数或业务参数中的字段可能会动态增加，组签名报文时，请勿固定字段名，应根据具体报文采用动态排序。

## 3、报文格式

### 提交数据

采用 HTTPS 标准的 POST 协议，为了保证接收方接收数据正确，传输数据必须签名。

### 3.1、请求报文

请求报文分为 3 层，分别为：

公共参数：所有接口通用， {}

业务参数：根据接口不同， biz\_content

业务扩展参数：根据特定场景扩展， biz\_content -> extend

请求示例：

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "329986",
  "biz_content": {
    "extend": {
      "terminal_no": "10300632",
      "auth_code": "134711323868398975"
    },
    "time_expire": "20201207154516",
    "out_trade_no": "N020201207144516370661",
    "time_start": "20201207144516",
    "total_amount": "1",
    "currency": "156",
    "client_ip": "113.71.218.142",
    "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
    "notify_url": "",
    "body": "test",
    "trans_type": "BSC"
  }
}
```

```

    },
    "mer_id": "YW0014406000000",
    "format": "json",
    "version": "1.0",
    "sign_type": "RSA2",
    "timestamp": "20201207144542"
}

```

## 3.2、响应报文

响应报文分为 3 层，分别为：

公共参数：所有接口通用， {}

业务参数：根据接口不同， response

业务扩展参数：根据特定场景扩展， response -> extend

正确返回数据：

```

{
  "response": {
    "sub_code": "ok",
    "sub_msg": "请求成功",
    "total_amount": 128,
  },
  "code": "20000",
  "msg": "ACQ.SUCCESS",
  "sign": "ERITJKEIJKJHKKKKKKKHJEREEEEEEEEEE"
}

```

业务级错误返回：

```

{
  "response": {
    "sub_code": "isp.unknown-error",
    "sub_msg": "系统繁忙"
  },
  "code": "20000",
  "msg": "Service Currently Unavailable",
  "sign": "ERITJKEIJKJHKKKKKKKHJEREEEEEEEEEE"
}

```

## 4、补单机制

注意：对后台通知交互模式，如果平台收到商户的应答不是纯字符串 success 或超过 5 秒后返回时，平台认为通知失败，平台会通过一定的策略（通知频率为 0/15/15/30/180/1800/1800/1800/1800/3600，单位：秒）间接性重新发起通知，尽可能提高通知的成功率，但平台不保证通知最终能成功。由于存在重新发送后台通知的情况，因此同样的通知可能会多次发送给商户系统。

商户系统必须能够正确处理重复的通知。平台推荐的做法是，当收到通知进行处理时，首先检查对应业务

数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回纯字符串 success。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重复插入数据造成的数据混乱。

## 5、错误码

返回码 code>20000 表示业务执行失败，msg 表示错误描述，具体的错误原因查看明细返回码和明细返回码描述；

code=20000 表示业务处理成功，交易结果请查看各接口返回值。

### 5.1、公共错误码

网关会对开发者的接口非业务调用错误做统一处理，返回码如下：

code (返回码)	msg (返回码描述)	sub_code (明细返回码)	sub_msg (明细返回码描述)	解决方案
20000	接口调用成功，调用结果请参考具体的所对应的业务返回参数。			
40000	缺少必选参数	missing-version	缺少版本参数	检查请求参数，缺少 version 参数
		missing-mer-id	缺少 mer_id 参数	检查请求参数，缺少 mer_id 参数
		missing-sign-type	缺少签名类型参数	检查请求参数，缺少 sign_type 参数
		missing-sign	缺少签名参数	检查请求参数，缺少 sign 参数
		missing-timestamp	缺少时间戳参数	检查请求参数，缺少 timestamp 参数
		missing-nonce-str	缺少随机数	检查请求参数，缺少 nonce_str 参数
		missing-biz-content	缺少业务参数集	检查请求参数，缺少 biz_content 参数
40002	非法的参数	invalid-api	无效 API	请检查 API 地址和请求方式是否正确，所有接口仅支持 POST

				方式提交
		invalid-version	版本号无效	检查入参 version, 当前版本 1.0
		invalid-format	无效的数据格式	检查入参 format, 目前只支持 json
		invalid-charset	字符集错误	请求参数 charset 错误, 目前支持格式: UTF-8
		invalid-sign-type	无效的签名类型	检查入参 sign_type, 目前只支持 RSA2
		invalid-timestamp	非法的时间戳参数	时间戳参数 timestamp 非法, 请检查格式需要为 yyyyMMddHHmmss
		invalid-mer-id	无效的 mer_id 参数	检查入参 mer_id, mer_id 不存在, 或者未上线
		invalid-merchant-disabled	商户状态不可用	请联系相关对接人员
		missing-sign-key	没有配置商户公钥	请联系相关对接人员
		missing-biz-content	缺少业务参数集错误	检查请求参数, 缺少 biz_content 参数
		invalid-sign	无效签名	1.公私钥是否是一对 2.检查公钥上传是否与私钥匹配 3.签名算法是否无误 4.签名结果要转 base64
40001	授权权限不足	invalid-merchant	无效商户	请检查商户号是否正确, 如确认无误, 请联系相关对接人员核实
40004	无效请求	invalid-request	无效请求	检查接口地址、请求方式是否正确
50000	业务处理失败	对应业务错误码, 明细错误码和解决方案请参见具体的 API		

50003	服务不可用	unknow-error	服务暂不可用(业务系统不可用)	稍后重试
		gateway-error	服务暂不可用(网关自身的未知错误)	稍后重试
		channel-error	渠道异常	稍后重试

注意: code 和 sub\_code、msg 和 sub\_msg 有层级关系, 但接口响应的数据会同时返回 4 个字段。

## 5.2、业务错误码

请查看具体业务接口。

# 6、请求参数

## 6.1、公共请求参数

参数名	类型	是否必填	最大长度	描述	示例
version	string	是	3	调用的接口版本, 固定为: 1.0	1.0
app_id	string	否	32	应用 ID, 多商户时使用	133927243
chnl_code	string	是	6	渠道标识	银联条码:000001
mer_no	string	是	32	渠道商户号	QRA00000001
format	string	否	5	请求参数格式, 仅支持 json	json
charset	string	否	10	字符集, 缺省为 UTF-8	UTF-8
encrypt_type	string	否	5	加密方式, 仅支持 AES	AES
timestamp	string	是	14	交易发生时间戳, yyyyMMddHHmmss 格式	
nonce_str	string	是	32	随机字符串	
sign_type	string	是	5	签名类型, 仅支持 MD5	MD5
sign	string	是	344		
biz_content	string	是	-	业务参数, 具体查看各【 <a href="#">API 接口</a> 】, {}	

				如需加密, 将 biz_conetnt 的值加密后转 base64	
--	--	--	--	-----------------------------------	--

## 6.2、公共响应参数

参数名	类型	是否必填	最大长度	描述	示例
code	string	是	10	网关返回码	40001
msg	string	是	64	网关返回码描述	
timestamp	string	否	14	交易发生时间戳, yyyyMMddHHmmss 格式	
nonce_str	string	否	16		
sign	string	是	344		
response	string	否	-	业务响应参数集, 如需加密, 将 respons 的值加密后转 base64	{ "sub_code" :" ", sub_msg" :" ", ... }

## 公共业务响应参数

参数名	类型	是否必填	最大长度	描述	示例
sub_code	string	是	-	业务返回码, 参见具体的 API 接 口文档	ACQ.INVALID_A MOUNT
sub_msg	string	是	-	业务返回码描述, 参见具体的 API 接口文档	无效金额
...				其它业务参数	

注意: 业务响应参数中的必填项, 通常是指请求成功 (code=20000) 状态下, 例如: 当签名校验失败时, 业务响应参数可能为空。



## 7、API 接口

### 7.1、统一下单

#### 应用场景

除付款码支付场景以外，商户系统先调用该接口在统一支付平台生成预支付交易单，返回正确的预支付交易会话标识后再按 Native、JSAPI、APP 等不同场景生成交易串调起支付。

#### 接口链接

URL 地址: /pay/unifiedorder

#### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
trans_type	string	是	16	交易方式, 详情见【 <a href="#">附录一</a> 】对内渠道	wx_app、 alipay_app
out_trade_no	string	是	64	商户系统内部订单号, 必须唯一	
total_amount	string	是	12	订单总金额, 单位: 分	
body	string	否	255	对一笔交易的具体描述信息。如果是多种商品, 请将商品描述字符串累加传给 body。	iPhone 11, 256G, 土豪金
attach	string	否	255	附加数据, 在查询 API 和支付通知中原样返回, 该字段主要用于商户携带订单的自定义数据。 <b>注意: 超过长度会被截断</b>	
currency	string	否	5	货币类型, 156 人民币	156
time_start	string	否	14	商户订单生成时间, 格式为 yyyyMMddHHmmss, 如 2009 年 12 月 25 日 9 点 10 分 10 秒表示为 20091225091010。	20091225091010
time_expire	string	否	14	订单失效时间, 格式为 yyyyMMddHHmmss, 如 2009	20091225091010

				年 12 月 27 日 9 点 10 分 10 秒表示为 20091227091010。	
notify_url	string	否	256	接收支付平台异步通知回调地址, 通知 url 必须为直接可访问的 url, 不能携带参数。	
client_ip	string	否	19	客户端 IP	100.12.10.5
extend	json string	否	-	特定渠道发起交易时需要的额外参数, 具体请查看【 <a href="#">附录二</a> 】各渠道请求参数说明	

## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	是	64	商户系统内部订单号, 必须唯一	
trade_no	string	是	64	平台订单号	
extend	json string	否	-	根据 trans_type 返回特定参数, 详情请查看【 <a href="#">附录三</a> 】	
<b>以下字段仅当 trans_type=bsc 时有效</b>					
trade_state	string	是	16	交易状态: SUCCESS--支付成功 REFUNDED--退款成功 REFUNDING--退款中 NOTPAY--未支付 CLOSED--已关闭 USERPAYING--用户支付中 (等待输密码, 需要查询) PAYERROR--支付失败(其他原因)	
total_amount	string	是	12	订单总金额, 单位: 分	total_amount
real_amount	string	是	12	实付金额, 单位: 分	real_amount
payment_type	string	否	1	支付方式 0: 未知 1: 建行龙支付 7: 云闪付 8: 微信支付	

				9: 支付宝	
currency	string	否	5	货币类型, 156 人民币	156
body	string	否	255	对一笔交易的具体描述信息。如果是多种商品, 请将商品描述字符串累加传给 body。	iPhone 11, 256G, 土豪金
out_transaction_id	string	否	64	第三方订单号 (云闪付/微信/支付宝等)	9551600056195246 8914253966530027
attach	string	否	255	附加数据, 在查询 API 和支付通知中原样返回, 该字段主要用于商户携带订单的自定义数据	

## 业务状态码

错误码	错误描述	解决方案
ACQ.SUCCESS	订单创建成功	支付结果请查看 <b>trade_state</b>
ACQ.SYSTEM_ERROR	接口返回错误	请立即调用查询订单 API, 查询当前订单的状态, 并根据订单状态决定下一步的操作, 如果多次调用依然报此错误码, 请联系客户经理。
ACQ.INVALID_PARAMETER	参数无效	检查请求参数是否完整、类型是否正确、长度是否正确, 修改后重新发起请求
ACQ.TOTAL_FEE_EXCEED	订单总金额超过限额	修改订单金额再发起请求
ACQ.CONTEXT_INCONSISTENT	交易信息被篡改	更换商家订单号后, 重新发起请求
ACQ.TRADE_HAS_SUCCESS	交易已被支付	确认该笔交易信息是否为当前买家的, 如果是则认为交易付款成功, 如果不是则更换商家订单号后, 重新发起请求
ACQ.TRADE_HAS_CLOSE	交易已经关闭	更换商家订单号后, 重新发起请求
ACQ.CHANNEL_TIMEOUT	渠道返回超时	trans_type=bsc 时, 请发起订单查询请求

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "329986",
  "biz_content": {
```

```

    "extend": {
      "terminal_no": "10300632",
      "auth_code": "134711323868398975"
    },
    "time_expire": "20201207154516",
    "out_trade_no": "NO20201207144516370661",
    "time_start": "20201207144516",
    "total_amount": "1",
    "currency": "156",
    "client_ip": "113.71.218.142",
    "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
    "notify_url": "",
    "body": "test",
    "trans_type": "BSC"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "LwkmZf4S827ssMpHp07gmTGnE6GzFMTY2/zDwie7MrY0AAJAMNYVwqBaVrAecRrhothiZf6mYrwIYthx7aU4Ic+kcCBTbrVokAPfxk4VHB0Jn15oYPWgFWwXVC2XHKwDAMYH4wFYCC2vuPZEAQ4973fgOp0+7Ys08X/DjqT8bj0iuKjqtUi0a50UUq5ZAD4PeooJsmLlLE3eHEXH9iDK93HHvHaPhiHzJkLkClEqjGcUAuAYIqxeaXoYE13Bd0SkjBsXQLvGY4UBwxRr/g6gMZE0B6vy06kzB6p+YjAw9qh10igqyZ3F3dEaiAK5P1nnsNrtvsEZSLobjpYX99i+Q==",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20201207144542"
}

```

## 响应示例

```

{
  "msg": "OK",
  "code": "20000",
  "sign": "U1u1u5lvwslp3Qs1/Oc+nUWjE5Nq4DY/Thc9m1811J0DqfZHz+Dt9o8uoLTayBNHTtPXvkyPGCjKGxCl5rULXde5kFa4uqp8GPeyylXEdHdQU1o0/w8cG0A6YUVuxXAW1xQhrAHpr4KA/sVGm86Vd1j7Mrz47imJUYZgW1sP17pYyI92GFA1Cew9qV2Frt5doyUYcwtV/8snsu138MvhKG+8ae0bBpo71PwDfoIihMTN++32BZ1ncfCxC2b9ra31hdybD56TW1dRktOLB0HRaDz1lQig2YNJncRkuUiZLamqBOCc2b1cmxoV5+z8C4c+9Atk3bhg5Xg0Qe2tLtgMjQ==",
  "response": {
    "payment_type": "8",
    "sub_code": "ACQ.SUCCESS",
    "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
    "sub_msg": "客户正在支付中，请等待",
    "trade_type": "BSC",
    "out_trade_no": "NO20201207144516370661",
    "trade_state": "USERPAYING",
    "real_amount": "0",
  }
}

```

```

    "trade_no": "YW0014406000000201207144542808747389",
    "total_amount": "1"
  }
}

```

## 7.2、查询订单

### 应用场景

该接口提供所有支付订单的查询, 商户可以通过查询订单接口主动查询订单状态, 完成下一步的业务逻辑。  
需要调用查询接口的情况:

- 1、◆ 当商户后台、网络、服务器等出现异常, 商户系统最终未接收到支付通知;
- 2、◆ 调用支付接口后, 返回系统错误或未知交易状态情况;
- 3、◆ 调用付款码支付 API, 返回 USERPAYING 的状态;
- 4、◆ 调用关单或撤销接口 API 之前, 需确认支付状态;

### 接口链接

URL 地址: /pay/orderquery

### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	二选一	64	商户系统内部订单号, 和平台订单号二选一	
trade_no	string		64	平台订单号, 和商户订单号二选一, 优先级: trade_no > out_trade_no	

### 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	是	64	商户系统内部订单号, 必须唯	

				—	
trade_no	string	是	64	平台订单号	
trade_state	string	是	16	交易状态: SUCCESS--支付成功 REFUNDED--退款成功 REFUNDING--退款中 NOTPAY--未支付 CLOSED--已关闭 USERPAYING--用户支付中 (等待输密码, 需要查询) PAYERROR--支付失败(其他原因)	
total_amount	string	是	12	订单总金额, 单位: 分	total_amount
real_amount	string	是	12	实付金额, 单位: 分	real_amount
payment_type	string	否	1	支付方式 0: 未知 1: 建行龙支付 7: 云闪付 8: 微信支付 9: 支付宝	
currency	string	否	5	货币类型, 156 人民币	156
body	string	否	256	对一笔交易的具体描述信息。如果是多种商品, 请将商品描述字符串累加传给 body。	iPhone 11, 256G, 土豪金
attach	string	否	256	附加数据, 在查询 API 和支付通知中原样返回, 该字段主要用于商户携带订单的自定义数据	
out_transaction_id	string	否	64	第三方订单号 (云闪付/微信/支付宝等)	9551600056195246 8914253966530027
extend	json string	否	-	根据 trans_type 返回特定参数, 详情请查看【 <a href="#">附录三</a> 】	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	查询成功	支付结果请查看 <code>trade_state</code>
ACQ.INVALID_PARAMETER	参数无效	检查请求参数，修改后重新发起请求
ACQ.TRADE_NOT_EXIST	要查询的交易不存在	检查传入的交易号是否正确，修改后重新发起请求
ACQ.TRADE_HAS_CLOSE	交易已经关闭	重新下单
ACQ.SYSTEM_ERROR	接口返回错误	重新发起请求
ACQ.CHANNEL_TIMEOUT	渠道返回超时	<code>trans_type=bsc</code> 时，请发起订单查询请求

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "371036",
  "biz_content": {
    "out_trade_no": "N020201207144516370661"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "PLRsLvW94v9q8p/tFYKVaIgzfts/Vgc8SZgLGkf7hdb5SQe19eYtBXmqanA7hPQZSY37/ooTJaRp2202ZHUIaLneGRTA
sH3BNmWkpSFUwrE60Zg0G0BEoJ3M3Ykgq4r4YPrpFEgu5aGvR8oRhwdSfHepWmk6nNp9v07m7NPviwdEXBeb0q/WatDHCi/j7Z/feYiL8
/zOV7vXvpFmJd+RQTnXpFVH1kQTZSseX5bPI/VKwvnXNnsRDkMhjyn5X+6RoazxJpdaiPwkIPH5MW8u+tIKR5aS8wEi9RiwAYNvfpgQnG
0agJBo6boYxZQ6skZZEEGtVNHVsSmBc9Qnxj7rmQ==",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20201207144844"
}
```

## 响应示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "N8hMV66PBPVHtFvqXT5Z2jKa47EgBlxTfM5pEiSausNC+hff1TeWH7TQVtdvBrWPYuCimEeV1n2wG8I2YEM0avz7dNv
/3E7dVqZ7Cc6k6nRQv1TNooprIATAKuuQRx1eHW3bxvJQ+bdWyu8D6b8HE6ZF82C+rRv3GVWDVA9bZGQE7rGVXAevUP+t08+9FE0j0/SH
```

```
ywNcVvkz4sJoKy07oMDswSkTWfheZ0w0gAi6ggSJxTUfowPko1fEePyVLlgr0YB+QHgjRKSwaoZ0q0E5n8+qqQpt2qismGf7z/vF+rycp
MQd58htMik8NTcr/r1jw1X/ZyXCAGnR3b9WHbMAA==",
    "response": {
        "payment_type": "8",
        "sub_code": "ACQ.SUCCESS",
        "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
        "sub_msg": "请求成功",
        "trade_type": "BSC",
        "out_trade_no": "NO20201207144516370661",
        "trade_state": "SUCCESS",
        "real_amount": "1",
        "trade_no": "YW001440600000201207144542808747389",
        "total_amount": "1"
    }
}
```

### 7.3、撤销订单

#### 应用场景

支付交易返回失败或支付系统超时，调用该接口撤销交易。如果此订单用户支付失败，系统会将此订单关闭；如果用户支付成功，系统会将此订单资金退还给用户。

注意：5 分钟以内的交易单可调用撤销，其他正常支付的单如需实现相同功能请调用申请退款 API。提交支付交易后调用【[查询订单 API](#)】，没有明确的支付结果再调用【[撤销订单 API](#)】。

调用支付接口后请勿立即调用撤销订单 API，建议支付后至少 15s 后再调用撤销订单接口。

#### 接口链接

URL 地址: /pay/reverse

#### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	二选一	64	商户系统内部订单号，要求 64 个字符内，只能是数字、大小写字母_ - *@ ，且在同一个商户号下唯一。 trade_no、out_trade_no 二选一，如果同时存在优先级：	



				trade_no > out_trade_no	
trade_no	string		64	平台订单号	

## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	是	64	商户系统内部订单号，必须唯一	
trade_no	string	是	64	平台订单号	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	撤销成功	
ACQ.INVALID_PARAMETER	参数无效	检查请求参数，修改后重新发起请求
ACQ.TRADE_NOT_EXIST	要撤销的交易不存在	检查传入的交易号是否正确，修改后重新发起请求
ACQ.TRADE_STATUS_ERROR	交易状态不合法	检查当前交易的状态是不是等待付款，只有等待付款状态下才能发起交易关闭。
ACQ.SYSTEM_ERROR	接口返回错误	重新发起请求

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "975595",
  "biz_content": {
    "trade_no": "YW00144060000006101053566445844"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "Q6dif6QTVIwqSj+OPTEnkIMlKcU15DrEXsYvp4sVlOBKrfL++eQKTZic6llJl3ftDuDBBjcdp1/WmM1/cur+MbJjkIM2pIE0Xb+NSF6UStmRgzQGhKfPb+x/lHiykNkZjEu55wrqd8508KuTwZcjQ5HRIqGotUx3eqsv9rmX+T7+b2iHcU13NWYCPu24NMhtqKKPR7Vqpph1CH84u5Iyh/Nn9gbq9HJRpTN1+qSyI85qsZ887I8AFD3ZwH0my8Vsf9YE9mp3kK8uYTh3RdRfXjDel8UmCee4aAwT/iXefLvZWlOkf7DOhiAiJsZ0fA6Q4yguzRP3IwVXOyy3u/QkSg==",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20210610105408"
}
```

}

## 响应示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "EzcltgeIW4gg2Pg+Q2yoVro31N8J6Du1r/wkoHEebbad3uuksdflP+/cU1pkruUGKzcoBCNXe5btLB5dBWh
VU12+cB8OD44OJDKWx5HHF1DHoKs+U3TLtmMm24kfPxTdc/RovOA1mRb82Z4hzSeqHHgE6a18F1KD6PEtPGi0W/VmIlyQ1p/
4MdnL1mySql+T4mtgEH+5opad9i5KR1LARQL+bQ7MOY18gSq+VDBX/xVarm17zGz1Vmky03G+SKaPT6z19emnaQx4pTVVQUE
sfZeocz+aJvuE8bGHkb0G37+rZ1oH74xAR/I5/bDUv2kUfrRcXnPpIboort77Rjc7MQ==",
  "response": {
    "sub_code": "ACQ.SUCCESS",
    "out_trade_no": "NO20210610105350523730",
    "sub_msg": "请求成功",
    "trade_no": "YW001440600000006101053566445844"
  }
}
```

## 7.4、关闭订单

### 应用场景

以下情况需要调用关单接口：商户订单支付失败需要生成新单号重新发起支付，要对原订单号调用关单，避免重复支付；系统下单后，用户支付超时，系统退出不再受理，避免用户继续，请调用关单接口。

**注意：订单生成后不能马上调用关单接口，最短调用时间间隔为 5 分钟。**

### 接口链接

URL 地址: /pay/closeorder

### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	二选一	64	商户系统内部订单号，要求 64 个字符内，只能是数字、大小写字	

				母_   * @ , 且在同一个商户号下唯一。 trade_no、out_trade_no 二选一, 如果同时存在优先级: trade_no > out_trade_no	
trade_no	string		64	平台订单号	
extend	string	否	-	特定渠道发起交易时需要的额外参数, 具体看各渠道请求参数说明	

## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	是	64	商户系统内部订单号, 必须唯一	
trade_no	string	是	64	平台订单号	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	订单关闭成功	
ACQ.INVALID_PARAMETER	参数无效	检查请求参数, 修改后重新发起请求
ACQ.TRADE_NOT_EXIST	要关闭的交易不存在	检查传入的交易号是否正确, 修改后重新发起请求
ACQ.TRADE_STATUS_ERROR	交易状态不合法	检查当前交易的状态是不是等待付款, 只有等待付款状态下才能发起交易关闭。
ACQ.SYSTEM_ERROR	接口返回错误	重新发起请求

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "434643",
  "biz_content": {
    "out_trade_no": "N020201207144516370661"
  },
  "mer_id": "YW0014406000000",
}
```

```

    "format": "json",
    "sign": "ftoxlsayjc+gGGATI2KhbEzF9JNzgl7cQ46Ym3CFN9D8bHT7W+kWnLMmjAtsbjVIC028QxoAHZrQLQgzn0My3/E1yRDsqYzH5mRt8hFunvyKyy2BGUPzy8l61bFbCem8kXHgoj5iWRSfIb7usQ10RuC9LLfQSUfC9AAgUFp2Xu2WfXLKeCdVytckFWprEeZeSW/mjzPlcWR2N9TrLYt00oEFF3ofQUDRGZ/13mbdmUPff1bPjYJ4+7H/ACA9K0kdSFMVgUbBjy2V0X7MU9tr17tCRlea6R0NqyL4Q3UKeoxZXPszppNDcOSXfZisdboLHl2DYymMHT11Gcvk9XTQ==",
    "version": "1.0",
    "sign_type": "RSA2",
    "timestamp": "20201207145135"
}

```

## 响应示例

```

{
  "msg": "业务处理失败",
  "code": "50000",
  "sign": "Cr5pViWTX2wYucKUnZDaxMJKCMWvkVFKW4IECsYgyQ7KX9FgPKouyeEmr5sdBxM1nCOvj+X1w8j098KBTnGIUPcoj6iL92xs1Z+FSdy0Ukxqg96v1T5yk1NVjMbqsMMpNtFbYforGeug47b/kzFvAHow4dup0GjEaqUL0JxaY9VUQ4wJppEZwnX9fbRAdCoDNLomLG0om+A6soG7+r/y/i79H+M/N4DNXdG3oNch/3y30P6Nawhc8kIYOTqRkKLTkEkzQ0/+XvIqC7QSeVEFj5H/YI5f0IsEtwKqqKDZdtzHe3BwajYohKp1gFNnowDwwkrt5DyQtLvYiAuu00qCg==",
  "response": {
    "sub_code": "ACQ.TRADE_STATUS_ERROR",
    "sub_msg": "交易状态不合法"
  }
}

```

## 7.5、申请退款

### 应用场景

当交易发生之后一段时间内，由于买家或者卖家的原因需要退款时，卖家可以通过退款接口将支付款退还给买家，平台将在收到退款请求并且验证成功之后，按照退款规则将支付款按原路退到买家帐号上。

注意：

- 1、有效退款时间根据各渠道而定
- 2、退款支持单笔交易分多次退款，多次退款需要提交原支付订单的商户订单号和设置不同的退款单号。申请退款总金额不能超过订单金额。**一笔退款失败后重新提交，请不要更换退款单号，请使用原商户退款单号**
- 3、每个支付订单的部分退款次数不能超过 50 次
- 4、如果同一个用户有多笔退款，建议分不同批次进行退款，避免并发退款导致退款失败
- 5、有优惠的订单不支持部分退款

## 接口地址

URL 地址: /pay/refund

## 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_refund_no	string	是	64	商户系统内部的退款单号, 商户系统内部唯一, 只能是数字、大小写字母_-*@, 同一退款单号多次请求只退一笔。	1217752501201 4070332333680 18
out_trade_no	string	二选一	64	商户系统内部订单号, 要求 64 个字符内, 只能是数字、大小写字母_-*@, 且在同一个商户号下唯一。 trade_no、out_trade_no 二选一, 如果同时存在优先级: trade_no > out_trade_no	
trade_no	string		50	平台订单号	
refund_amount	string	是	12	退款金额, 单位分	
refund_reason	string	否	256	退款的原因说明	正常退款
currency	string	否	5	货币类型, 可选, 默认为原交易的货币类型	156
notify_url	string	否	256	接收统一支付平台异步通知回调地址, 通知 url 必须为直接可访问的 url, 不能携带参数。	
extend	string	否	-	特定渠道发起交易时需要的额外参数, 具体看各渠道请求参数说明	

## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_refund_no	string	是	64	商户系统内部的退款单号, 商户系统内部唯一, 只能是数字、	1217752501201 4070332333680

				大小写字母_-*@ , 同一退款单号多次请求只退一笔。	18
refund_no	string	是	64	平台退款单号	
out_trade_no	string	是	64	商户系统内部订单号, 要求 64 个字符内, 只能是数字、大小写字母_-*@ , 且在同一个商户号下唯一。 trade_no、out_trade_no 二选一, 如果同时存在优先级: trade_no > out_trade_no	
trade_no	string	是	64	下单接口返回的平台订单号	
refund_amount	string	是	12	退款金额, 单位: 分	
real_refund_amount	string	是	12	实际退款金额, 单位: 分	
refund_reason	String	否	256	退款的原因说明	正常退款
refund_currency	string	否	5	货币类型, 可选, 默认为原交易的货币类型	156
refund_state	string	是	16	退款状态: SUCCESS—退款成功 REFUNDCLOSE—退款关闭 PROCESSING—退款处理中 CHANGE—退款异常	
extend	string	否	-	特定渠道发起交易时需要的额外参数, 具体看各渠道响应参数说明	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	退款订单创建成功	退款结果请查看 refund_state
ACQ.SYSTEM_ERROR	系统错误	请使用相同的参数再次调用
ACQ.INVALID_PARAMETER	参数无效	请求参数有错, 重新检查请求后, 再调用退款
ACQ.REFUND_FEE_ERROR	交易退款金额有误	请检查传入的退款金额是否正确
ACQ.REFUND_FEE_EXCEED	退款金额超限	检查退款金额是否超过付款金额, 重新

		修改请求后，重新发起退款
ACQ.TRADE_NO_REPEAT	退款单号重复	使用该单号请求退款查询接口
ACQ.TRADE_NOT_EXIST	交易不存在	检查请求中的交易号和商户订单号是否正确，确认后重新发起
ACQ.TRADE_HAS_CLOSE	交易已关闭	该交易已关闭，不能再进行退款，请确认请求退款的交易是否未支付或者已完成退款
ACQ.TRADE_NOT_ALLOW_REFUND	当前交易不允许退款	检查当前交易的状态是否为交易成功状态以及签约的退款属性是否允许退款，确认后，重新发起请求

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "964312",
  "biz_content": {
    "out_trade_no": "N020201207144516370661",
    "out_refund_no": "N020201207145531918708",
    "refund_amount": "1",
    "currency": "156",
    "notify_url": "http://"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "N0SkMuAWrYFqaf1bm6FGyz2pzoRfsmL8j0C9ZNpTpGMLHvTFAqUII1FFI41ZuezG9K61MFsyKyxf9EF3DjH3Rj9fxv+E4FthNVlaHqqeKVq7HCtvEgBmlaaLLD66+K05H0yce52y204T1FLbkns0y3r8miycpdQzxI3ibLhf+WLOMaVCN/POAoXZoT/ppZHnFqbIFxjaRqRffUp1mlHWI6jwb+lp0oGmAeLI2DQpNZjUc9hKgHhQ9D/mgQhpNOp1m7W8yLqde6VUubVKJp0ZdACZ8HzchMS2KSThf3JQbmPfBAi5a6FXdGUqEihsav5Pv19d499/KQ4FN/ekc15Pqkg==",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20201207145548"
}
```

## 响应示例

```
{
  "msg": "OK",
  "code": "20000",
}
```

```

"sign": "d0cVWF0m9gTPLFCTuya1Z84TuRZZgUwQ1ukwdCzIfwaiuNIqaq1SxF2cXaiuYGjJG52XjoR6MHa9xUR69QIIB77ERyG7
+XpTPPHAai693C+UK1UrM50y15EuCathOG3wL5NUAS9y+xTysQH2utntvh10Dco5wLn+O/KGv0d3QHZFnRbq7N0/QaWozcuvzabSHLQxX
nnVm0q+uesA1ZqCiebaKQq+VbGKkhkut4YBfibGVgt1Vb2r1gR1xdYZqXBM101IDNr0k0u24uwPI5a+O9UQ3RwbJ1uT61qWksU+NWAPgR
UkJz6Pi00iH5qFQZaw+yjfYZd5bvP0vUAUxY6okg==",
"response": {
  "sub_code": "ACQ.SUCCESS",
  "real_refund_amount": "1",
  "out_refund_no": "NO20201207145531918708",
  "out_trade_no": "NO20201207144516370661",
  "refund_amount": "1",
  "sub_msg": "请求成功",
  "refund_state": "SUCCESS",
  "refund_no": "YW001440600000201207145548216464048",
  "trade_no": "YW001440600000201207144542808747389",
  "total_amount": "1"
}
}

```

## 7.6、退款查询

### 应用场景

提交退款申请后，通过调用该接口查询退款状态。退款有一定延时，退款到账时间根据渠道而定。

### 接口地址

URL 地址: /pay/refundquery

### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_refund_no	string	二选一	64	商户系统内部的退款单号，商户系统内部唯一，只能是数字、大小写字母_- *@，同一退款单号多次请求只退一笔。	12177525012 01407033233 368018
refund_no	string		64	平台退款订单号，优先级： refund_no > out_refund_no	



## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_refund_no	string	是	64	商户系统内部的退款单号, 商户系统内部唯一, 只能是数字、大小写字母_-*@, 同一退款单号多次请求只退一笔。	12177525012 01407033233 368018
refund_no	string	是	64	平台退款单号	
out_trade_no	string	是	64	商户系统内部订单号, 要求32个字符内, 只能是数字、大小写字母_-*@, 且在同一个商户号下唯一。	
trade_no	string	是	64	平台交易单号	
total_amount	string	是	12	该笔退款所对应的交易的订单金额, 单位: 分	
refund_amount	string	是	12	退款总金额, 单位: 分	
real_refund_amount	string	是	12	实际退款金额, 单位: 分	
refund_state	string	是	12	退款状态: SUCCESS—退款成功 REFUNDCLOSE—退款关闭 PROCESSING—退款处理中 CHANGE—退款异常	
extend	string	否	-	特定渠道发起交易时需要的额外参数, 具体看各渠道响应参数说明	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	查询成功	退款结果请查看 refund_state
ACQ.SYSTEM_ERROR	系统错误	请使用相同的参数再次调用
ACQ.INVALID_PARAMETER	参数无效	请求参数有错, 重新检查请求后, 再调用退款

ACQ.TRADE_NOT_EXIST	查询退款的交易不存在	检查请求中的交易号和商户订单号是否正确，确认后重新发起
---------------------	------------	-----------------------------

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "090845",
  "biz_content": {
    "out_refund_no": "N020201207145531918708"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "a13oZZKHa0cp+YGv7W3MbfSXOPmeqE54f0zdIfG3eu/Xe70b7KRoxro6Q42PSSEHJJIf3ZdA+c2eQcWGLytWGenwsfjFMSlinM1BWM/J5WVdu7neufhesvKYKg6ceQGcf+L7uXuZi+b7QUPjdQP2peuoKprZV9VfcedHCpQeTt0jCYdVDVXoca37HgBTdam4VzuvE VSA4KCJVODBD6zfwKeLNfkb/UMdKMVwuSf1RuEndeshoqJ3BD9ZOV1cfuZtTf7kjMZJyYLjYqWiggaHsSsXeJPr41w5HDynioFukrL0mj77yTGG7TPHFTabjqsntHcJZwxIeg4iKTempV5vA=",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20201207145931"
}
```

## 响应示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "d0cVWF0m9gTPLFCTuyalZ84TuRZZgUwQ1ukwdCzIfwaiuNIqaq1SxF2cXaiuYGjJG52XjoR6MHa9xUR69QIIB77ERYG7+XpTPPHAai693C+UK1UrM50y15EuCathOG3wL5NUAS9y+xTysQH2utntvh10Dco5wLn+O/KGv0d3QHZFnRbq7N0/QaWozcuvzabSHLQxXnnVm0q+uesA1ZqCiebaKQq+VbGKkhkut4YBfibGVgtlvb2r1grLxdYZqXBMl0lIDNr0k0u24uwPI5a+O9UQ3RwbJ1uT61qWksU+NWAPgR UkJz6Pi00iH5qFQZaw+yjFYZd5bvP0vUAUxY6okg=",
  "response": {
    "sub_code": "ACQ.SUCCESS",
    "real_refund_amount": "1",
    "out_refund_no": "N020201207145531918708",
    "out_trade_no": "N020201207144516370661",
    "refund_amount": "1",
    "sub_msg": "请求成功",
    "refund_state": "SUCCESS",
    "refund_no": "YW0014406000000201207145548216464048",
    "trade_no": "YW0014406000000201207144542808747389",
  }
}
```

```

    "total_amount": "1"
  }
}

```

## 7.7、退款查询扩展（根据支付单号查询）

### 应用场景

提交退款申请后，通过调用该接口查询退款状态。退款有一定延时，退款到账时间根据渠道而定。  
分页查询

### 分页查询

当一个订单部分退款超过 10 笔后，商户用平台订单号或商户订单号调退款查询 API 查询退款时，默认返回前 10 笔和 total\_refund\_count（订单总退款次数）。商户需要查询同一订单下超过 10 笔的退款单时，可传入订单号及 offset 来查询，平台会返回 offset 及后面的 10 笔，以此类推。当商户传入的 offset 超过 total\_refund\_count，则系统会返回报错 PARAM\_ERROR。

举例：

一笔订单下的退款单有 36 笔，当商户想查询第 25 笔时，可传入订单号及 offset=24，平台会返回第 25 笔到第 35 笔的退款单信息，或商户可直接传入退款单号查询退款

### 接口地址

URL 地址：/pay/refundqueryext

### 请求参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	二选一	64	商户系统内部的订单号，商户系统内部唯一，只能是数字、大小写字母 _- *@，同一退款单号多次请求只退一笔。	12177525012 01407033233 368018
trade_no	string		64	平台订单号，优先级：trade_no > out_trade_no	

offset	int	否	2	偏移量, 当部分退款次数超过 10 次时可使用, 表示返回的查询结果从这个偏移量开始取记录, 默认为 0	0
--------	-----	---	---	------------------------------------------------------	---

## 响应参数

参数名	类型	是否必填	最大长度	描述	示例
out_trade_no	string	是	64	商户系统内部订单号, 要求 32 个字符内, 只能是数字、大小写字母_!*@, 且在同一个商户号下唯一。	
trade_no	string	是	64	平台交易单号	
total_amount	string	是	12	原订单交易金额, 单位: 分	
+refund_list	json array	是			[[{}], {}, {}]
└ out_refund_no	string	是	64	商户系统内部的退款单号, 商户系统内部唯一, 同一退款单号多次请求只退一笔。	
└ refund_no	string	是	64	平台退款单号	
└ refund_amount	string	是	12	申请退款金额, 单位: 分	
└ real_refund_amount	string	是	12	实际退款金额, 单位: 分	
└ refund_state	string	是	12	退款状态: SUCCESS—退款成功 REFUNDCLOSE—退款关闭 PROCESSING—退款处理中 CHANGE—退款异常	
└ refund_state_des	string	是	100	退款状态描述	

## 业务错误码

返回码	描述	解决方案
ACQ.SUCCESS	查询成功	退款结果请查看 refund_state
ACQ.SYSTEM_ERROR	系统错误	请使用相同的参数再次调用

ACQ.INVALID_PARAMETER	参数无效	请求参数有错，重新检查请求后，再调用退款
ACQ.TRADE_NOT_EXIST	查询退款的交易不存在	检查请求中的交易号和商户订单号是否正确，确认后重新发起

## 请求示例

```
{
  "charset": "UTF-8",
  "encrypt_type": "aes",
  "nonce_str": "508598",
  "biz_content": {
    "out_trade_no": "NO20201207144516370661",
    "offset": "0"
  },
  "mer_id": "YW0014406000000",
  "format": "json",
  "sign": "BGsJ04JsjjRIheBdvGck9jW0GB39gIL5yTNJY1bXesUxc32NQY6z0fwwk+FfuwQ26ttfnmdDGWduYaXiw8Jf1B3Mcz1xfqqunaEhNDdk+e4Tj0QkXB0BXbvUufw4hw7Xs8+4HPy7fWba6xwDaS/HtbzRGJOPna1tJtjVjrBIXnGcT+HzVWDP5BQ/YSAV01BAyAug1EMI6Ez3p8QvcplFz148FCaaQ9a/QwyrsoKxBPVfDPWEuxRK+oDmf+R6BR3kKxne9RMpacz5VTZPX0CeNUyyo7MR01ea9e0Ir+RI+7bwzZGMYhcUMuNkzwmFx5q+GUxdV4kXfG0CJ3vnuApg==",
  "version": "1.0",
  "sign_type": "RSA2",
  "timestamp": "20201207151622"
}
```

## 响应示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "CQiYWezBmm2wmX81Wjyk6IYMeoCEB910C0yCO/ID0VF0MX+zwYX9qYCNFxCtBdFDqJMgx+Z686jZQThSBzBlH/cIAXpv2H7v0BR70DYC76vm19BnrRxUff71YfYr1c/OpdSaiFab7y51C6nFmKx8qApRWmir9mjp7pzg5dUYxqaT2yhrNJGygLZJ+YxskbeS/idHbT236YsdHokVXYMxwGFYTFMPhgCRMs02ymq+byURivCT4+wR8IbwCLgWRTjM6LLe+rr2NnSG3kjDeqyEVD17eFDJzuIFPHWg/+DwsL4ouhDVJgZLqCya5ENscBuMFRW/E01/m5u0USfqZpVuQg==",
  "response": {
    "sub_code": "ACQ.SUCCESS",
    "out_trade_no": "NO20201207144516370661",
    "sub_msg": "请求成功",
    "refund_count": 1,
    "refund_list": [{
```

```
        "out_refund_no": "NO20201207145531918708",
        "refund_no": "YW0014406000000201207145548216464048",
        "refund_amount": "1",
        "real_refund_amount": "0",
        "refund_state": "SUCCESS",
        "refund_state_des": "受理成功"
    }],
    "trade_no": "YW0014406000000201207144542808747389",
    "total_amount": "1"
}
}
```

## 7.8、交易结果通知

### 应用场景

支付完成后，平台会把相关支付结果及用户信息通过数据流的形式发送给商户，商户需要接收处理，并按文档规范返回应答。

#### 注意：

- 1、同样的通知可能会多次发送给商户系统。商户系统必须能够正确处理重复的通知。
- 2、后台通知交互时，如果平台收到商户的应答不符合规范或超时，平台会判定本次通知失败，重新发送通知，直到成功为止（在通知一直不成功的情况下，平台总共会发起多次通知，通知频率为 15s/15s/30s/3m/10m/20m/30m/30m/30m/60m/3h/3h/3h/6h/6h - 总计 24h4m），但平台不保证通知最终一定能成功。
- 3、在订单状态不明或者没有收到平台结果通知的情况下，建议商户主动调用平台【[查询订单 API](#)】和【[退款查询 API](#)】确认订单状态。

#### 特别提醒：

- 1、商户系统对于支付结果通知的内容一定要做**签名验证,并校验返回的订单金额是否与商户侧的订单金额一致**，防止数据泄漏导致出现“假通知”，造成资金损失。
- 2、当收到通知进行处理时，首先检查对应业务数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回结果成功。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

## 接口链接

该链接是通过【[统一下单 API](#)】和【[申请退款 API](#)】中提交的参数 notify\_url 设置，如果链接无法访问，商户将无法接收到平台通知。

通知 url 必须为直接可访问的 url，不能携带参数。示例：notify\_url: "https://xxxx.com/pay/notify"

## 请求参数

请求参数同各业务接口的响应参数；

例如：

支付结果通知 同 [【查询订单返回结果】](#)

退款结果通知 同 [【退款查询返回结果】](#)

## 响应参数

商户接收到通知后，返回字符串【SUCCESS】

## 支付结果通知示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "N8hMV66BPVHtFvqXT5Z2jKa47EgB1xXTfM5pEiSausNC+hff1TeWH7TQVtdvBrWPYuCimEeV1n2wG8I2YEMOavz7dNv/3E7dVqZ7Ccfk6nRQv1TNooprIATAKuuQRx1eHW3bxvJQ+bDWyu8D6b8HE6ZF82C+rrV3GVWDVA9bZGQE7rGVXAevUP+t08+9FE0j0/SHyWncVkz4sJJoKy0oMDswSkTWFheZ0w0gAi6ggSjxTUfoWPko1fEEPyVLlgrR0YB+QHGjRKSwaoZ0q0E5n8+qqQpt2qismGf7z/vF+rycpMQd58htMik8NTcr/r1jw1X/ZyXCagnR3b9WHbMAA==",
  "response": {
    "payment_type": "8",
    "sub_code": "ACQ.SUCCESS",
    "attach": "aaano=xxxxxxxxxxxxx,bbbno=xxxxxxxxxxxxx",
    "sub_msg": "请求成功",
    "trade_type": "BSC",
    "out_trade_no": "N020201207144516370661",
    "trade_state": "SUCCESS",
    "real_amount": "1",
    "trade_no": "YW0014406000000201207144542808747389",
    "total_amount": "1"
  }
}
```

## 退款结果通知示例

```
{
  "msg": "OK",
  "code": "20000",
  "sign": "d0cVWF0m9gTPLFCTuyalZ84TurZZgUwQ1ukwdCzIfwaiuNIqaq1SxF2cXaiuYGjJG52XjoR6MHa9xUR69QIIB77ERyG7
+XpTPPHAai693C+UK1UrM50y15EuCath0G3wL5NUAS9y+xTysQH2utntvh10Dco5wLn+0/KGv0d3QHZFnRbq7N0/QaWozcuvzabSHLQxX
nnVm0q+uesA1ZqCiebaKQq+VbGKkhkut4YBfibGVgtlVb2r1gRlxdYZqXBM10lIDNr0k0u24uwPI5a+O9UQ3RwbJ1uT61qWksU+NWAPgR
UkJz6Pi00iH5qFQZaw+yjfYZd5bvP0vUAUxY6okg==",
  "response": {
    "sub_code": "ACQ.SUCCESS",
    "real_refund_amount": "1",
    "out_refund_no": "NO20201207145531918708",
    "out_trade_no": "NO20201207144516370661",
    "refund_amount": "1",
    "sub_msg": "请求成功",
    "refund_state": "SUCCESS",
    "refund_no": "YW001440600000201207145548216464048",
    "trade_no": "YW001440600000201207144542808747389",
    "total_amount": "1"
  }
}
```

## 附录一：交易类型说明

交易类型编码	交易类型	支付场景	对应支付产品
csb	主扫支付	用户扫商家的码	
bsc	被扫支付	商家扫用户的码	
wx_app	微信 App 支付	APP	APP
wx_mp	微信 JSAPI 支付	MP, 微信内置浏览器	JSAPI
wx_applet	微信小程序支付	APPLET	JSAPI

## 附录二：各渠道请求扩展参数说明

wx\_mp:

字段名	类型	是否必填	最大长度	描述	示例
-----	----	------	------	----	----



app_id	string	是	64	公众号/小程序的 appid	
open_id	string	是	64	用户在微信公众号分配的唯一标识	
device_info	string	否	32	自定义参数, 可以为终端设备号(门店号或收银设备 ID), PC 网页或公众号内支付可以传 "WEB"	
receipt	string	否	8	Y, 传入 Y 时, 支付成功消息和支付详情页将出现开票入口。需要在微信支付商户平台或微信公众平台开通电子发票功能, 传此字段才可生效	
limit_pay	string	否	32	上传此参数 no_credit--可限制用户不能使用信用卡支付	

## wx\_applet:

字段名	类型	是否必填	最大长度	描述	示例
app_id	string	是	64	公众号/小程序的 appid	
open_id	string	是	64	用户在微信公众号分配的唯一标识	
device_info	string	否	32	自定义参数, 可以为终端设备号(门店号或收银设备 ID), PC 网页或公众号内支付可以传 "WEB"	
receipt	string	否	8	Y, 传入 Y 时, 支付成功消息和支付详情页将出现开票入口。需要在微信支付商户平台或微信公众平台开通电子发票功能, 传此字段才可生效	
limit_pay	string	否	32	上传此参数 no_credit--可限制用户不能使用信用卡支付	

## bsc:

字段名	类型	是否必填	最大长度	描述	示例
auth_code	string	是	30	支付授权码	13123456789123 456789

terminal_no	string	是	10	终端号	90812345
device_no	string	是	32	设备编号	xxxxxxxxx
device_sn	string	否	6	终端流水	000001
longitude	string	否	10	经度	22.8888
latitude	string	否	10	纬度	103.8888

## 附录三：各渠道响应扩展参数说明

csb:

字段名	类型	是否必填	最大长度	描述	示例
code_url	string	是	300	此 url 用于生成支付二维码, 然后提供给用户进行扫码支付。 注意: code_url 的值并非固定, 使用时按照 URL 格式转成二维码即可	http://

wx\_app/wx\_mp/wx\_applet (以下字段同微信支付对应产品的响应参数必填字段):

字段名	类型	是否必填	最大长度	描述	示例
appId	string	是	16	公众号 id	wx88888888888888
timeStamp	string	是	32	时间戳	1414561699
nonceStr	string	是	32	随机字符串	5K8264ILTKCH16 CQ2502SI8ZNMT M67VS
package	string	是	128	订单详情扩展字符串	prepay_id=12345 6789
signType	string	是	32	签名方式	MD5
paySign		是	64	签名	C380BEC2BFD727 A4B6845133519F 3AD6

参考微信内 H5 调起支付: [https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7\\_7&index=6](https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=7_7&index=6)

wx\_h5 (以下字段同微信支付 H5 的响应参数必填字段):

字段名	类型	是否必填	最大长度	描述	示例
appid	string	是	32	调用接口提交的应用 ID	wx88888888888888

					8888
mch_id	string	是	32	调用接口提交的商户号	1900000109
nonce_str	string	是	32	微信返回的随机字符串	5K8264ILTKCH16 CQ2502SI8ZNMT M67VS
sign	string	是	64	微信返回的签名	
prepay_id	string	是	64	微信生成的预支付回话标识, 用于后续接口调用中使用, 该值有效期为 2 小时	wx201410272009 395522657a6903 89285100
trade_type	string	是	16	调用接口提交的交易类型, 取值如下: JSAPI, NATIVE, APP, 详细说明见	JSAPI
mweb_url	string	是	300	<b>mweb_url 为拉起微信支付收银台的中间页面, 可通过访问该 url 来拉起微信客户端, 完成支付,mweb_url 的有效期为 5 分钟。</b>	https://wx.tenpay.com/cgi-bin/mmpayweb-bin/checkmweb?prepay_id=wx2016121516420242444321ca0631331346&package=1405458241