

	BestPay MPI 文档
	用户手册 – V2.8

---

# BestPay 网关接入规范

## 付款码收银（条码支付）

User Guide

V2.8

---

**Bestpay MAPI 用户手册 V2.8**

---

© 2017，天翼电子商务有限公司

本文档所有内容，未经本公司事前书面授权，不得以任何方式(包括储存于光盘或任何储存媒体内) 作全部或局部之翻印、仿制或转载

# 1 版本更新记录

日期	版本	说明	作者
2016-05-23	V1.4	正式发行版本	黄银海
2016-05-17	V1.5	增加 5.2：支付返回增加优惠金额字段 增加 6.2：查询返回增加优惠金额字段	刘永富
2016-06-15	V1.6	增加 6.4：查询退款说明 增加 8.5：部分退款说明	刘永富/黄银海
2016-07-1	V1.7	修改部分话术	刘永富
2016-07-07	V1.8	增加 BE110078 返回错误码	刘永富
2016-08-10	V1.9	增加 7.3：支付返回增加商户营销优惠成本字段 增加 8.2：查询返回增加商户营销优惠成本字段	闫嘉丽
2016-09-08	V2.0	交易 key 名称统一，查询修正退款字段	黄银海
2016-11-25	V2.1	撤销接口成功响应报文修改，新增退款回调 9	黄银海
2016-12-30	V2.2	付款码请求流水号需要使用条码退款业务，必须为偶数位；更新分账说明 11.1；更新商户 key 说明和接入说明	黄银海
2017-01-06	V2.3	更新 md5 mac 说明描述	黄银海
2017-01-11	V2.4	更新 8.4 和 10.3 的示例，新增 11.2 通知接口（后台）中有 BANKID 字段解释	黄银海
2017-01-23	V2.5	更新 11.2 通知接口（后台）中有 BANKID 字段解释，更新 10.2,10.4 说明;更新 5.2,5.3 应答示例和说明；更新 6.2,6.3 应答示例和说明；更新 8.3,8.4 说明和示例	黄银海
2017-03-21	V2.6	新增 5，请求响应终端号和自定义终端号字段	黄银海



BestPay MPI 文档

用户手册 – V2.8

2017-03-28	V2.7	更新 5，更新请求响应终端号和自定义终端号字段长度	黄银海
2017-04-26	V2.8	更新 5.2 请求参数说明 orderDate，增加 24 小时制时间说明	黄银海

## 2 目录

1	版本更新记录 .....	2
2	目录 .....	4
3	MAPI 简介 .....	6
4	商户接入说明 .....	7
5	付款码支付 .....	8
5.1	业务流程 .....	8
5.2	请求说明 .....	9
5.3	应答说明 .....	11
5.4	请求及响应示例 .....	12
5.5	接口说明 .....	13
5.6	客户端测试用例 .....	15
5.7	错误码 .....	18
6	交易查询 .....	20
6.1	请求说明 .....	20
6.2	应答说明 .....	21
6.3	请求及响应示例 .....	23
6.4	接口说明 .....	23
6.5	客户端测试用例 .....	24

6.6	错误码.....	27
<b>7</b>	<b>通知接口（异步）.....</b>	<b>28</b>
<b>8</b>	<b>退款接口.....</b>	<b>31</b>
8.1	业务流程.....	31
8.2	退款请求说明.....	31
8.3	退款应答说明.....	33
8.4	请求响应示例.....	34
8.5	接口说明.....	34
8.6	客户端测试用例.....	35
8.7	错误码.....	38
<b>9</b>	<b>退款回调通知接口.....</b>	<b>39</b>
<b>10</b>	<b>撤销交易接口.....</b>	<b>41</b>
10.1	请求说明.....	41
10.2	应答说明.....	42
10.3	响应示例.....	43
10.4	接口说明.....	43
10.5	客户端测试用例.....	44
10.6	错误码.....	46
<b>11</b>	<b>附录说明.....</b>	<b>47</b>
11.1	交易支付-分账说明.....	47
11.2	通知接口（后台）中有 BANKID 字段解释.....	48

## 3 MAPI 简介

BestPay MAPI 提供了便于商户系统实现自有系统与 BestPay 网关平台的接入服务。

## 4 商户接入说明

### 1. 申请翼支付的正式商户号，商户数据 KEY，商户交易 KEY

下图中的 **key** 为数据 **KEY**（数据 key 为一串随机数字字母）：

```
MERCHANTID=987654321&ORDERSEQ=20060314000001&ORDERDATE=20060314150908
```

```
&ORDERAMOUNT=1000&CLIENTIP=127.0.0.1&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D4343E4548C02
```

下图中的 **key** 为交易 key 为 6 位数字：

MERCHANTPWD	签约商户密码	6	M
-------------	--------	---	---

### 2. 需要使用快捷支付的商户，商户需要申请商户的银行快捷支付权限

3. 需要退款和冲正的商户，商户需要申请退款和冲正的权限，并且提供服务器的固定 IP 以便配置访问白名单

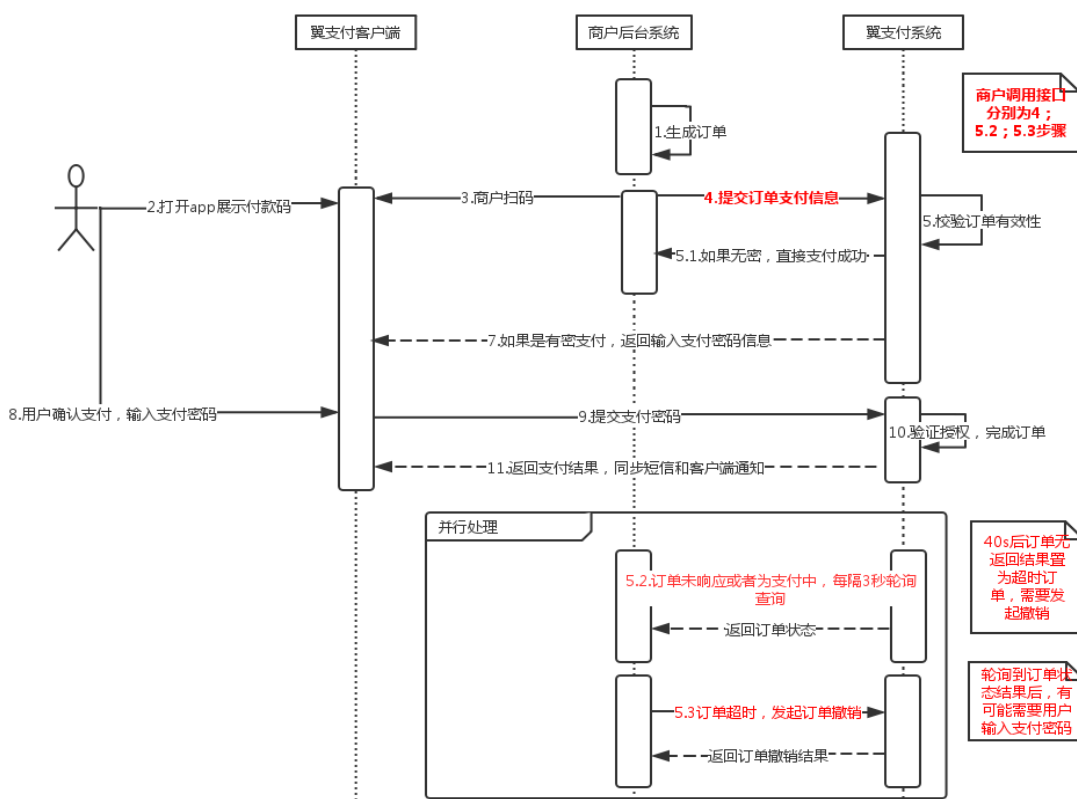
### 4. 需要分账支付的商户，商户需要申请分账交易权限

5. 字段必填 M/O 的说明：M 为必填，O 为可选

6. 商户需提供门店信息。目的让用户客户端正确显示历史支付信息

## 5 付款码支付

### 5.1 业务流程



备注：

- 1) 步骤 4，统一订单接口：详见 [3.2](#)
- 2) 步骤 5.2，订单查询接口：详见 [4.1](#)
- 3) 步骤 5.3，订单撤销接口：详见 [7.1](#)



## 5.2 请求说明

### 1. 请求地址

环境	HTTP请求地址
正式环境	<a href="https://webpaywg.bestpay.com.cn/barcode/placeOrder">https://webpaywg.bestpay.com.cn/barcode/placeOrder</a>

### 2. 请求参数

HTTP表单提交				
字段	名称	类型长度	必填	说明
merchantId	商户号	30	M	由翼支付网关平台统一分配
subMerchantId	子商户号	30	O	由商户平台自己分配
barcode	条形码号	30	M	商户POS扫描用户客户端条形码
orderNo	订单号	30	M	由商户平台提供，支持纯数字、纯字母、字母+数字组成， <b>全局唯一（如果需要使用条码退款业务，订单号必须为偶数位）</b>
orderReqNo	订单请求交易流水号	30	M	由商户平台提供，支持纯数字、纯字母、字母+数字组成， <b>与订单号一致（如果需要使用条码退款业务，流水号必须为偶数位）</b>
channel	渠道	5	M	默认填：05
busiType	业务类型	5	M	默认填：0000001
orderDate	订单日期	20	M	由商户提供，长度14位，格式yyyyMMddhhmmss (说明： <b>该时间必须为24小时制</b> )

orderAmt	订单总金额	10	M	单位：分。订单总金额 = 产品金额+附加金额
productAmt	产品金额	10	M	单位：分
attachAmt	附加金额	10	M	单位：分
goodsName	商品名称	256	O	商品信息
storeId	门店号	10	M	门店号
backUrl	后台返回地址	255	O	商户提供的用于异步接收交易返回结果的后台url，若不需要后台返回，可不填， <b>若需要后台返回，请保障地址可用</b>
ledgerDetail	分账信息	256	O	商户需要在结算时进行分账情况，需填写此字段，详情见接口说明 <a href="#">分账明细</a>
attach	附加信息	128	O	商户附加信息
mac	MAC校验域	256	M	采用标准的MD5算法，由商户实现， <b>MD5加密获得32位大写字符</b>
mchntTmNum	商户自定义终端号	50	O	商户自定义终端号
deviceTmNum	设备终端号	50	O	设备终端号

### 3.mac 说明

MAC 校验域说明：MD5: 顺序（注意是字段名是大写）

MERCHANTID=123456789&ORDERNO=123456&ORDERREQNO=1234567&

ORDERDATE=200603141210&BARCODE=111111&ORDERAMT=1&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D4343E4548C02

## 5.3 应答说明

### HTTP 接口响应 (json格式)

字段	result字段说明	名称	类型 长度	必填	说明
success		是否成功	5	M	true : 成功。 false : 失败。 (代表商户下单的成功与失败, <b>不代表订单的支付状态</b> )
<b>result (说明当 success为true时, result为下单详情, success为false时, result为null)</b>	merchantId	商户号	30	M	由翼支付网关平台统一分配
	orderNo	订单号	30	M	查询到的商户订单号
	orderReqNo	订单请求 流水号	30	M	查询到的商户请求流水号
	orderDate	预留字段		O	始终为null
	ourTransNo	翼支付交 易号	30	M	翼支付生成的内部流水号(用 户支付后生成)
	transAmt	订单金额	10	M	单位: 分
	transStatus	交易状态	1	M	A: 请求(支付中) B: 成功(支付成功) C: 失败( <b>订单状态结果</b> )
	encodeType	签名方式	1	O	1代表MD5; 3代表RSA; 9代表CA; <b>默认为1</b>
	sign	sign校验 域	30	M	十六进制
	coupon	优惠金额	10	M	订单优惠金额, 用户使用代金 券或立减的金额, 金额为分
	scValue	商户营 销 优惠成本	10	O	商户营销优惠成本
	payerAccount	付款人账 号	50	O	
	payeeAccount	收款人账 号	50	O	

	payChannel	付款明细	80	O	
	productDesc	备注	200	O	
	refundFlag	退款标示	1	O	
	customerId	客户登陆 账号	15	O	
	mchntTmNum	商户自定义 终端号	50	O	
	deviceTmNum	设备终端 号	50	O	
errorCode		错误码	10	O	当success为true时，为null， 当success为false时，为错误 码
errorMsg		错误描述	10	O	当success为true时，为null， 当success为false时，为错误 描述

注：sign 校验说明

sign 校验域说明 MD5: 顺序（注意是字段名是大写）

MERCHANTID=0018888888&ORDERNO=1234567890&

ORDERREQNO=20141001&ORDERDATE=null&OURTRANSNO=201508080808&

TRANSAMT=10&TRANSSTATUS=B&ENCODETYPE=1&KEY=344C4FB521F5A52EA28FB7FC79AE  
A889478D4343E4548C02。ORDERDATE 在下单接口返回时暂未使用，目前始终为 null（十六进制，  
再比较）

## 5.4 请求及响应示例

请求参数示例：

merchantId=043101180050000&subMerchantId=043101180050000&barcode=5156650028548

86972&orderNo=1433734609560&orderReqNo=14337346095601&orderDate=20150608113649&channel=05&busiType=0001&&orderAmt=1&productAmt=1&attachAmt=0&goodsName=%E6%9D%A1%E7%A0%81%E6%94%AF%E4%BB%98&storeId=201231&backUrl=http%3A%2F%2F127.0.0.1%3A8030%2FwebBgNotice.action&ledgerDetail=&attach=&mac=3B5E200FD1EF8988693F242438A32475

响应参数示例：

成功：{"success":true,"result":{"merchantId":"01440109025345000","orderNo":"20170123140956abcd","orderReqNo":"20170123140956000001","orderDate":null,"transStatus":"B","transAmt":"1","ourTransNo":"2017012300000923157471","encodeType":"1","sign":"C6E7A64F9828943D9DB6C7089F4B7F2D","payerAccount":null,"payeeAccount":null,"payChannel":null,"productDesc":null,"refundFlag":null,"customerId":null,"coupon":"0","scValue":"0","respCode":null,"respDesc":null},"errorCode":null,"errorMsg":null}

失败：{"success":false,"result":null,"errorCode":"BE300001","errorMsg":"订单 MAC 域验证失败"}

## 5.5 接口说明

1. 商户的商户号由翼支付统一分配，商户需与翼支付的运营人员对接，提供商户资料，申请正式的商户号。
2. 商户要保证订单号的唯一性。
3. 订单请求流水号不能重复,建议以日期时间（yyyyMMDDhhmmss 格式）加一固定长度（不小于 6 位）流水号组成，如：20061012132425 + 0000001 等。
4. 订单流水号和订单号可以相同。（使用翼支付提供的“流程测试”商户时除外，如果相同会提示订单或流水号重复的错误）
5. MAC 校验，保证了交易信息到翼支付网关平台的完整性，参与 MAC 运算的字符及其顺序如下：
  - a) 顺序

MERCHANTID=123456789&ORDERNO=123456&ORDERREQNO=1234567&ORDERDATE=200603141210&BARCODE=111111&ORDERAMT=1&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D4343E4548C02

**b) 此处的 KEY 为商户的数据 KEY，可以通过翼支付的统一运行地址生成。此数据 KEY 对本文档其它接口进行 MAC 校验时同样适用。 <https://managerwg.bestpay.com.cn/login.jsp>**

c) 要求：将上述固定顺序组织的字符串，参数名均为大写，使用标准 md5 算法进行摘要，摘要结果 16 进制字符串，字符串建议使用大写，就是校验域 MAC 的值，并在提交订单时，将 MAC 值和订单信息一起提交到翼支付网关翼支付网关平台。

d) 需保证 MAC 校验时中的各个参数，与下单请求时的参数完全一致。

6. 商户若需分账，则填写 ledgerDetail 字段，例如：

a) 规则：023101111:4|023102222:3|023103333:1

b) 说明：订单总金额 4+3+1=8 分,商户 023101111 分账 4 分，每组对应关系之间 “|” 分隔

c) 分账权限与一般支付权限不同。商户分账支付的权限需要额外申请。

**7. 下单支付接口返回的 success 值并不是最终支付结果。支付结果在下单成功情况下，result 的 transStatus 字段代表支付结果。**

8. 返回结果中的翼支付网关流水号 ourTransNo，当用户完成支付后才会生成。对于有密支付情况，由于用户未实际支付，返回的流水号为空字符串。商户可在查询接口获得该流水号。

9. 接口响应中的 sign 值是网关根据返回结果进行的前面，保证收到结果的完整性，商户可以根据需要进行验签，目前签名默认为 MAC 形式，参与 MAC 运算的字符及其顺序如下：

顺序：MERCHANTID=0018888888&ORDERNO=1234567890&  
ORDERREQNO=20141001&ORDERDATE=null&OURTRANSNO=201508080808&TRANSAM  
T=10&TRANSSTATUS=B&ENCODETYPE=1&KEY=123456

ORDERDATE 在下单接口返回时暂未使用，目前始终为 null。

## 5.6 客户端测试用例

注：此用例仅供参考

```
import com.bestpay.paycenter.commons.util.CryptTool;
import org.apache.http.Consts;
import org.apache.http.HttpEntity;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
import javax.net.ssl.SSLContext;
import javax.xml.namespace.QName;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.SSLContextBuilder;
import org.apache.http.conn.ssl.TrustStrategy;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
public class TestClient {
    public static void main(String[] args) throws Exception{
        String merchantId = "043101180050000";
        String orderNo = "1433734609522";
        String orderReqNo = "14337346095221";
        String orderDate = "20150608113649";
        String barcode = "512582454851204002";
        String orderAmt = "1";

        StringBuilder sb = new StringBuilder();//组装 mac 加密明文串
        sb.append("MERCHANTID=").append(merchantId);
        sb.append("&ORDERNO=").append(orderNo);
        sb.append("&ORDERREQNO=").append(orderReqNo);
        sb.append("&ORDERDATE=").append(orderDate);
        sb.append("&BARCODE=").append(barcode);
        sb.append("&ORDERAMT=").append(orderAmt);
        sb.append("&KEY=").append("111");//此处是商户的 key

        String mac = CryptTool.md5Digest(sb.toString());//进行 md5 加密(商户自己封装 MD5 加密工
具类，此处只提供参考)

        Map<String,String> param = new HashMap<String, String>();//组装请求参数

        param.put("merchantId", merchantId);
        param.put("subMerchantId", merchantId);
        param.put("barcode", barcode);
        param.put("orderNo", orderNo);
        param.put("orderReqNo", orderReqNo);
        param.put("orderDate", orderDate);
        param.put("channel", "05");
        param.put("busiType", "0001");
        param.put("TransType", "B");
        param.put("orderAmt", orderAmt);
        param.put("productAmt", "1");
```



```
param.put("attachAmt", "0");
param.put("goodsName", "条码支付");
param.put("storeId", "201231");
param.put("backUrl", "http://127.0.0.1:8030/webBgNotice.action");
param.put("ledgerDetail", "");
param.put("attach", "");
param.put("mac", mac);
```

```
//创建信任证书
```

```
CloseableHttpClient httpClient = createSSLClientDefault();
HttpPost httpPost = null;
CloseableHttpResponse response = null;
try {
    httpPost = new HttpPost("https://webpaywg.bestpay.com.cn/barcode/placeOrder");
    List<NameValuePair> paramList = new ArrayList<NameValuePair>();
    for (String key : param.keySet()) {
        paramList.add(new BasicNameValuePair(key, param.get(key)));
    }
    httpPost.setEntity(new UrlEncodedFormEntity(paramList, Consts.UTF_8));
    httpPost.setConfig(RequestConfig.custom().setConnectTimeout(30000).setSocketTimeou
t(30000).build());
    response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    int statusCode = response.getStatusLine().getStatusCode();
    if(HttpStatus.SC_OK == statusCode)//如果响应码是 200
        System.out.println(EntityUtils.toString(entity));
} finally {
    if (response != null) {
        response.close();
    }
    if (httpPost != null) {
        httpPost.releaseConnection();
    }
    httpClient.close();
}
```

```
}  
  
//创建链接  
public static CloseableHttpClient createSSLClientDefault() throws Exception {  
    SSLContext sslContext = new SSLContextBuilder().loadTrustMaterial(null, new AllTrustStrategy()).build();  
    SSLConnectionSocketFactory sslSf = new SSLConnectionSocketFactory(sslContext);  
    return HttpClients.custom().setSSLSocketFactory(sslSf).build();  
}  
  
//加载证书  
private static class AllTrustStrategy implements TrustStrategy {  
  
    public boolean isTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException {  
        return true;  
    }  
}  
}
```

## 说明

当使用 org.apache.http.client 之外的第三方类封装 HTTP 请求时，需要注意

1. HTTP 请求中的字符串，需要经过 URL 编码，其中的中文字符使用 UTF-8。
2. HTTP 请求中的 Content-Type 使用 application/x-www-form-urlencoded; charset=UTF-8，请求方法为 POST。

## 5.7 错误码

错误码	描述	备注
BE199999	请求参数有误	检查 HTTP 请求相关代码
BARCODE_VALID	条形码验证异常	
ATE_ERROR		
849	卡信息解密失败	用户信息异常
5146	查询绑卡信息出错	用户信息异常
4008	绑卡信息不存在	用户信息异常

-301	订单 MAC 域验证失败	
-302	校验订单出错	检查订单金额
-304	商户未配置此交易权限	
1001	商户未注册	
1002	商户未配置密钥信息	检查数据 KEY 是否申请
1003	商户未开通银行	
-3061	分账商户不存在	
-3062	分账金额有误	
-3063	分账金额总和不等于订单金额	
-3064	分账商户父商户有误	
-3065	分账信息不允许有两个相同商 户	
BARCODE_STOR	条码门店号不能为空	
E_CODE_IS_EMPTY		
Y		
BE999999	系统繁忙，请稍后再试	系统原因，联系技术支协查

## 6 交易查询

### 6.1 请求说明

#### 1. 请求地址

环境	HTTPS请求地址
正式环境	<a href="https://webpaywg.bestpay.com.cn/query/queryOrder">https://webpaywg.bestpay.com.cn/query/queryOrder</a>

#### 2. 请求参数

HTTP表单提交				
字段	名称	类型 长度	必填	说明
merchantId	商户号	30	M	由翼支付网关平台统一分配
orderNo	订单号	30	M	由商户平台提供，支持纯数字、纯字母、字母+数字组成
orderReqNo	订单请求交易流水号	30	M	由商户平台提供，支持纯数字、纯字母、字母+数字组成
orderDate	订单日期	30	M	日期格式：yyyyMMdd
mac	mac校验域	30	M	默认为0，当加密方式为1时有意义，采用标准的MD5算法。MD5加密获得32位大写字符

#### 3. mac 说明

MAC 校验域说明：MD5: 顺序（注意是字段名是大写）

MERCHANTID=0018888888&ORDERNO=1234567890&

ORDERREQNO=20141001&ORDERDATE=201507020&KEY=344C4FB521F5A52EA28FB7FC79AE  
A889478D4343E4548C02

## 6.2 应答说明

HTTP 接口响应（json格式）

字段	result字段说明	名称	类型	必填	说明
success		是否查询到订单	5	M	True：成功。False：失败。代表商户下单的成功与失败， <b>不代表订单的支付状态</b>
result（说明当 success 为 true 时，result 为下单详情，success 为 false 时，result 为 null）	merchantId	商户号	30	M	由翼支付网关平台统一分配
	orderNo	商户总订单号	30	M	查询到的商户订单号
	orderReqNo	商户总订单请求流水号	30	M	查询到的商户请求流水号
	orderDate	商户下单时间	20	M	yyyyMMddHHmmss，查询到的订单实际日期，可能与查询请求日期不同
	ourTransNo	网关平台流水号	30	M	翼支付生成的内部流水号
	transAmt	交易金额	10	M	单位： <b>分</b>
	transStatus	交易状态	1	M	A：请求（支付中）B：成功（支付成功）C：失败G：订单作废（ <b>订单状态结果</b> ）
	encodeTyp	加密方式	1	O	1 代表 MD5，3 代表 RSA，9 代表 CA，默认为 1

e				
sign	sign 校验域		M	验证签名，签名方式根据 encodeType 决定
refundFlag	退款标识	1	M	0 代表没有退款，1 已退款 2 部分退款 3 已冲正
customerID	客户支付手机号	15	M	客户条码消费时的支付手机号，中间 4 位被隐去，用*代替。如 180****2687
coupon	优惠金额	10	M	订单优惠金额，用户使用代金券或立减的金额，金额为分
scValue	商户营销优惠成本	10	O	商户营销优惠成本
payerAccount	付款人账号	50	O	
payeeAccount	收款人账号	50	O	
payChannel	付款明细	80	O	
productDesc	备注	200	O	
errorCode	错误码	10	O	当 success 为 false 时取此值，result 为空
errorMsg	错误描述	10	O	当 success 为 false 时取此值，result 为空

## sign 校验域说明

sign 校验域说明 MD5: 顺序（注意是字段名是大写）

MERCHANTID=0018888888&ORDERNO=1234567890&

ORDERREQNO=20141001&ORDERDATE=2015070215000000&OURTRANSNO=201508080808  
&TRANSAMT=10&TRANSSTATUS=B&ENCODETYPE=1&KEY=344C4FB521F5A52EA28FB7FC79A  
EA889478D4343E4548C02

## 6.3 请求及响应示例

### 请求示例

orderReqNo=20150713093607000001&orderNo=20150713093607&orderDate=20150713  
&mac=880D9E9DE657E3C351697D3B6CFE5626&merchantId=043101180050000

HTTP 请求对参数的顺序没有要求，保证参数名称完全一致（包括大小写）。

### 响应示例

订单存在

```
{"success":true,"result":{"merchantId":"01440109025345000","orderNo":"20170123140956abcd",
"orderReqNo":"20170123140956000001","orderDate":"20170123140956","transStatus":"B","tran
sAmt":"1","ourTransNo":"2017012300000923157471","encodeType":"1","sign":"13462C3ADECD
93D5DCC0815A0EDCCBAE","payerAccount":null,"payeeAccount":null,"payChannel":null,"produc
tDesc":null,"refundFlag":"1","customerId":"153****4685","coupon":"0","scValue":"0"},"errorCode"
:null,"errorMsg":null}
```

订单不存在

```
{"success":false,"result":null,"errorCode":"BE110062","errorMsg":"没有找到符合条件的记录。"}
```

其它错误

```
{"success":false,"result":null,"errorCode":"BE199999","errorMsg":"订单日期格式有误"}
```

接口响应的格式为 JSON。

## 6.4 接口说明

1. 可以查询支付或者退款交易：

a) 查询支付：商户订单号（orderNo）+ 商户总订单请求流水号（orderReqNo）

- b) 查询退款：orderNo 字段传送原扣款订单号 ( oldOrderNo ) + orderReqNo 字段传送 退款流水号 ( refundReqNo ) , [参数含义见退款请求参数](#)
2. MAC 校验，保证了交易信息到翼支付网关平台的完整性，参与 MAC 运算的字符及其顺序如下：
- a) 顺序：MERCHANTID=00188888888&ORDERNO=1234567890&  
ORDERREQNO=20141001&ORDERDATE=201507020&KEY=344C4FB521F5A52EA28FB7F  
C79AEA889478D4343E4548C02
- b) 要求：将上述固定顺序组织的字符串，参数名均为大写，使用标准 md5 算法进行摘要，摘要结果 16 进制字符串，字符串建议使用大写, 就是校验域 MAC 的值，并在提交订单时，将 MAC 值和订单信息一起提交到翼支付网关翼支付网关平台。KEY 值为商户的数据 KEY。
3. 接口响应的 success 字段标识的是查询过程的结果，并不代表交易的成功和失败。交易的结果当 success 为 true 时，取 result 中的 tranStatus 字段。字段为 A，代表请求状态，表明用户正在输入密码或交易本身在进行中。字段为 B 时，代表本次交易成功。字段为 C 时，代表本次交易失败。字段为 G 时，代表本次交易已被作废。
4. 订单请求时，为了加快查询的速度，需给定下单的日期，实际的查询范围，会在给定日期的前一天到给定日期的后一天范围内进行。
5. 接口响应中的 sign 值是网关根据返回结果进行的前面，保证收到结果的完整性，商户可以根据需要进行验签，目前签名默认为 MAC 形式，参与 MAC 运算的字符及其顺序如下：

顺序：MERCHANTID=00188888888&ORDERNO=1234567890&  
ORDERREQNO=20141001&ORDERDATE=2015070215000000&OURTRANSNO=201508080  
808&TRANSAMT=10&TRANSSTATUS=B&ENCODETYPE=1&KEY=344C4FB521F5A52EA28FB7  
FC79AEA889478D4343E4548C02

## 6.5 客户端测试用例

```
import com.bestpay.paycenter.commons.util.CryptTool;  
import org.apache.http.Consts;  
import org.apache.http.HttpEntity;  
import org.apache.http.HttpStatus;  
import org.apache.http.NameValuePair;
```



```
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
```

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.net.ssl.SSLContext;
import javax.xml.namespace.QName;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.SSLContextBuilder;
import org.apache.http.conn.ssl.TrustStrategy;
```

```
public class QueryClient {
    public static void main(String[] args) throws Exception {
        String merchantId = "043101180050000";
        String orderNo = "20150713093607";
        String orderReqNo = "20150713093607000001";
        String orderDate = "20150713";

        StringBuilder sb = new StringBuilder();//组装 mac 加密明文串
        sb.append("MERCHANTID=").append(merchantId);
        sb.append("&ORDERNO=").append(orderNo);
        sb.append("&ORDERREQNO=").append(orderReqNo);
        sb.append("&ORDERDATE=").append(orderDate);
        sb.append("&KEY=").append("111");//此处是商户的 key
```

```
        String mac = CryptTool.md5Digest(sb.toString());//进行 md5 加密(商户自己封装 MD5 加密工具类，此处只提供
        参考)
```

```
Map<String, String> param = new HashMap<String, String>();//组装请求参数，参数名大小写敏感
```

```
param.put("merchantId", merchantId);
param.put("orderNo", orderNo);
param.put("orderReqNo", orderReqNo);
param.put("orderDate", orderDate);
param.put("mac", mac);
```

```
//创建信任证书
```

```
CloseableHttpClient httpClient = createSSLClientDefault();
HttpPost httpPost = null;
CloseableHttpResponse response = null;
try {
    //发起 HTTP 的 POST 请求
    httpPost = new HttpPost("https://webpaywg.bestpay.com.cn/query/queryOrder");
    List<NameValuePair> paramList = new ArrayList<NameValuePair>();
    for (String key : param.keySet()) {
        paramList.add(new BasicNameValuePair(key, param.get(key)));
    }
    //UTF8+URL 编码
    httpPost.setEntity(new UrlEncodedFormEntity(paramList, Consts.UTF_8));
    httpPost.setConfig(RequestConfig.custom().setConnectTimeout(30000).setSocketTimeout(30000).build());
    response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    int statusCode = response.getStatusLine().getStatusCode();
    if (HttpStatus.SC_OK == statusCode)//如果响应码是 200
        System.out.println(EntityUtils.toString(entity));
} finally {
    if (response != null) {
        response.close();
    }
    if (httpPost != null) {
        httpPost.releaseConnection();
    }
    httpClient.close();
}
}

public static CloseableHttpClient createSSLClientDefault() throws Exception {
    SSLContext sslContext = new SSLContextBuilder().loadTrustMaterial(null, new AllTrustStrategy()).build();
```

```
SSLConnectionSocketFactory sslSf = new SSLConnectionSocketFactory(sslContext);  
return HttpClientBuilder.create().setSSLContext(sslSf).build();  
}  
//加载证书  
private static class AllTrustStrategy implements TrustStrategy {  
  
    public boolean isTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException {  
        return true;  
    }  
}  
}
```

## 6.6 错误码

错误码	描述	备注
1002	商户未配置密钥信息	检查数据 KEY 是否申请
BE300001	订单 MAC 域验证失败	
BE199999	请求参数有误	具体的参数错误，在描述中有体现
BE110062	没有找到符合条件的记录。	
BE999999	系统繁忙，请稍后再试	系统原因，联系技术支协查
BE110078	原交易不存在或超过退款限制时间	

## 7 通知接口（异步）

### 1. 通知接口说明

由商户提供，在下单接口的 backUrl 字段中配置

### 2. 参数

#### 异步后台通知（http）

字段	名称	类型长度	必填	说明
UPTRANSEQ	翼支付网关平台交易流水号	30	M	由翼支付网关平台提供，商户必须保存该信息，作为对帐依据
MERCHANTID	商户号	30	M	商户号
TRANDATE	翼支付网关平台交易日期	8	M	由翼支付网关平台提供，商户必须保存该信息，格式：yyyyMMDD，商户对账、清算报表以此日期为准
RETNCODE	处理结果码	4	M	由翼支付网关平台统一定义，商户需保存，作为对帐数据。结果码为“0000”表示支付成功，其他值则表示支付失败
RETNINFO	处理结果解释码	256	M	由翼支付网关平台统一定义，对支付结果的说明码
ORDERREQTRANSEQ	订单请求交易流水号	32	M	从商户发送的订单的信息中获得，翼支付网关平台原值传回
ORDERSEQ	订单号	32	M	从商户发送的订单的信息中获得，翼支付网关平台原值传回
ORDERAMOUNT	订单总金额	10	M	从商户发送的订单的信息中获得，翼支付网关平台原值传回，单位： <span style="color: red;">分</span> ，订单总金额 =

				产品金额+附加金额
PRODUCTAMOUNT	产品金额	10	M	从商户发送的订单的信息中获得，翼支付网关平台原值传回，单位：分
ATTACHAMOUNT	附加金额	10	M	从商户发送的订单的信息中获得，翼支付网关平台原值传回，单位：分
CURTYPE	币种	10	M	默认填 RMB
ENCODETYPE	加密方式	1	M	0：不加密;1：MD5摘要(默认)
BANKID	银行编码	64	M	支付使用的银行编码 <a href="#">详见11.2解释</a>
ATTACH	商户附加信息	128	O	从商户发送的订单的信息中获得，翼支付网关平台原值传回
UPREQTRANSEQ	网关平台请求银行流水号	32	C	从商户发送的订单的信息中获得，翼支付网关平台原值传回
UPBANKTRANSEQ	银行流水号	32	O	银行流水号
PRODUCTNO	产品号码	32	O	如充值会返回充值的号码。
SIGN	数字签名	256	M	数字签名算法由翼支付网关平台统一提供，作为核查依据（为1时有效）

### 3. Sign 验签域说明

sign 校验域说明 MD5: 顺序（注意是字段名是大写）

UPTRANSEQ=20080101000001&MERCHANTID=0250000001&ORDERID=2006050112564931556&

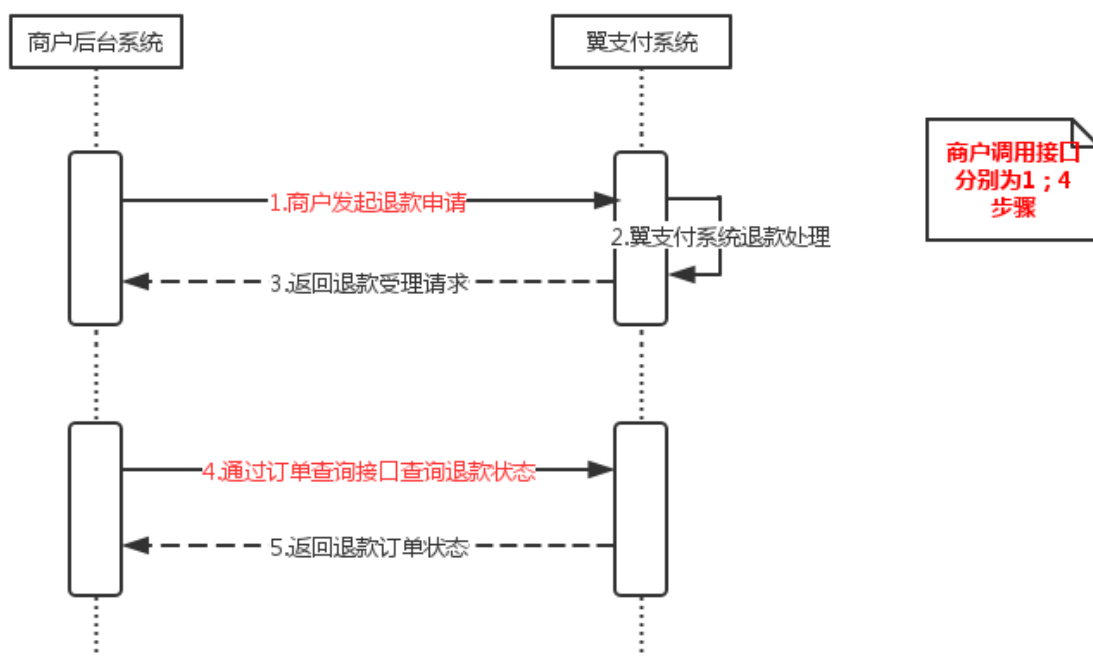
PAYMENT=10000&RETNCODE=0000&RETNINFO=0000&PAYDATE=20060101&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D4343E4548C02

### 4. 异步通知说明（重要）

1. 商户平台在收到后台支付结果后，请直接在应答时写入格式为 UPTRANSEQ\_XXXXXX 的字符串，其中 UPTRANSEQ\_ 为固定写死，XXXXXX 为翼支付网关平台发送过去的翼支付网关平台交易流水号。
2. 翼支付网关平台，如果在 60 秒钟没有收到应答，则会重复发送，重复通知时间为 1 分钟、3 分钟、5 分钟、1 小时、2 小时、4 小时、8 小时若仍没有收到应答将不再发送支付结果；翼支付网关平台如果收到应答，则不再发送支付结果。

## 8 退款接口

### 8.1 业务流程



### 8.2 退款请求说明

#### 1. 请求地址

环境	HTTPS请求地址
正式环境	<a href="https://webpaywg.bestpay.com.cn/refund/commonRefund">https://webpaywg.bestpay.com.cn/refund/commonRefund</a>

## 2. 请求参数

### HTTP表单提交

字段	名称	类型长度	必填	说明
merchantId	商户代码	20	M	由翼支付网关平台统一分配
subMerchantId	商户子代码	20	O	商户子代码，如没有则填空
merchantPwd	商户调用密码	20	M	商户执行时需填入相应密码， <b>又称：交易key</b>
oldOrderNo	原扣款订单号	30	M	原扣款成功的订单号
oldOrderReqNo	原订单请求支付流水号	30	M	原扣款成功的请求支付流水号
refundReqNo	退款流水号	30	M	该流水在商户处必须唯一。 <a href="#">若存在部分退款场景，具体见说明6.5</a>
refundReqDate	退款请求日期	14	M	yyyyMMDD
transAmt	退款交易金额	12	M	单位为 <b>分</b> ，小于等于原订单金额
ledgerDetail	分账明细	1024	O	商户需要在结算时进行分账情况，需填写此字段，详情见接口说明 <a href="#">分账明细</a>
channel	渠道	5	M	默认填05
mac	Mac校验域	32	M	<b>MD5加密获得32位大写字符</b>
bgUrl	退款回调地址	256	O	商户的退款回调地址，当退款受理

## 3. Mac 说明



MAC 校验域说明：MD5: 顺序（注意是字段名是大写）

MERCHANTID=123456789&MERCHANTPWD=123456&OLDORDERNO=123456&

OLDORDERREQNO=1234567&REFUNDREQNO=200603141210&REFUNDREQDATE=20150720  
&

TRANSAMT=1&LEDGERDETAIL=123456789:1&KEY=344C4FB521F5A52EA28FB7FC79AEA88947  
8D4343E4548C02

## 8.3 退款应答说明

HTTPs 接口响应（json格式）

字段	名称	类型长度	必填	说明
success	是否成功	5	M	true：成功，false：失败
result	调用返回值		O	结果：{ sign=, refundReqNo=, oldOrderNo=, transAmt=}; sign是签名串；refundReqNo退款请求流水号，oldOrderNo原订单号，transAmt交易金额，分为单位。
errorCode	错误码	10	O	当success为false时取此值，result为空
errorMsg	错误描述	10	O	当success为false时取此值，result为空

## 8.4 请求响应示例

请求示例

```
oldOrderReqNo=20150713093607000001&oldOrderNo=20150713093607&mac=880D9E9DE657E3C351697D3B6CFE5626&merchantId=043101180050000&transAmt=1&channel=01&refundReqNo=r20150713093607000001&refundReqDate=20150708&merchantPwd=123456
```

响应示例

成功

```
{"success":true,"result":{"sign":"A5FD5994035D2A69F68C1951485D4F6C","refundReqNo":"201611180000002","oldOrderNo":"20161118151028abcd","transAmt":"1"},"errorCode":null,"errorMsg":null}
```

失败

```
{"success":false,"result":null,"errorCode":"BE110062","errorMsg":"没有找到符合条件的记录。"}
```

## 8.5 接口说明

1. 商户要保证退款流水号的唯一性。
2. **部分退款**时，原扣款订单号（oldOrderNo）和原订单请求支付流水号（oldOrderReqNo）需一致，每笔部分退款的请求，退款流水号（refundReqNo）需唯一。

例如：订单金额为 100 分（oldOrderNo = 201606150001，oldOrderReqNo=201606150001），分两次退款；

第一笔：oldOrderNo=201606150001, oldOrderReqNo=201606150001, refundReqNo=**01**, transAmt=20;

第二笔为：oldOrderNo=201606150001, oldOrderReqNo=201606150001, refundReqNo=**02**, transAmt=80.

3. MAC 校验，保证了交易信息到翼支付网关平台的完整性，参与 MAC 运算的字符及其顺序如下：
  - a. MERCHANTID=123456789&MERCHANTPWD=123456&OLDORDERNO=123456&OLDORDE

RREQNO=1234567&REFUNDREQNO=200603141210&REFUNDREQDATE=20150720&TRAN  
SAMT=1&LEDGERDETAIL=123456789:1&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D  
4343E4548C02

- b. 商户业务平台将上述固定顺序组织的字符串，参数名均为大写，使用标准 md5 算法进行摘要，摘要结果 16 进制字符串，字符串建议使用大写，就是校验域 MAC 的值，将 MAC 值和订单信息调用接口时，一起提供给翼支付网关平台。

4. 对返回的 [8.3](#) 中 result 验签顺序：

OLDORDERNO=20170119114534659495&REFUNDREQNO=20170119144446990000&TRANSA  
MT=1&KEY=9A5357588EFB80D378403FF3041DCCCCC582C050E527D6

## 8.6 客户端测试用例

```
import com.bestpay.paycenter.commons.util.CryptTool;
import org.apache.http.Consts;
import org.apache.http.HttpEntity;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;
```

```
import javax.net.ssl.SSLContext;
import javax.xml.namespace.QName;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.HashMap;
```

```
import java.util.List;
import java.util.Map;
import org.apache.http.conn.ssl.SSLConnectionSocketFactory;
import org.apache.http.conn.ssl.SSLContextBuilder;
import org.apache.http.conn.ssl.TrustStrategy;

public class TestClient {
    public static void main(String[] args) throws Exception {
        String merchantId = "043101180050000";
        String merchantPwd = "111";
        String oldOrderNo = "1533734609531";
        String oldOrderReqNo = "15337346095231";
        String refundReqNo = "20150608113649";
        String refundReqDate = "20150720";
        String transAmt = "1";
        String channel = "02";
        String ledgerDetail = "043101180050000:1";

        StringBuilder sb = new StringBuilder();//组装 mac 加密明文串
        sb.append("MERCHANTID=").append(merchantId);
        sb.append("&MERCHANTPWD=").append(merchantPwd);
        sb.append("&OLDORDERNO=").append(oldOrderNo);
        sb.append("&OLDORDERREQNO=").append(oldOrderReqNo);
        sb.append("&REFUNDREQNO=").append(refundReqNo);
        sb.append("&REFUNDREQDATE=").append(refundReqDate);
        sb.append("&TRANSAMT=").append(transAmt);
        sb.append("&LEDGERDETAIL=").append(ledgerDetail);
        sb.append("&KEY=").append("111");//此处是商户的 key

        String mac = CryptTool.md5Digest(sb.toString());//进行 md5 加密(商户自己封装 MD5 加密工具类，此处只提供
参考)

        Map<String, String> param = new HashMap<String, String>();//组装请求参数

        param.put("merchantId", merchantId);
        param.put("merchantPwd", merchantPwd);
        param.put("oldOrderNo", oldOrderNo);
        param.put("oldOrderReqNo", oldOrderReqNo);
        param.put("refundReqNo", refundReqNo);
        param.put("refundReqDate", refundReqDate);
        param.put("transAmt", transAmt);
```

```
param.put("channel", channel);
param.put("ledgerDetail", ledgerDetail);
param.put("mac", mac);

//创建信任证书
CloseableHttpClient httpClient = createSSLClientDefault();
HttpPost httpPost = null;
CloseableHttpResponse response = null;
try {
    httpPost = new HttpPost("https://webpaywg.bestpay.com.cn/refund/commonRefund");
    List<NameValuePair> paramList = new ArrayList<NameValuePair>();
    for (String key : param.keySet()) {
        paramList.add(new BasicNameValuePair(key, param.get(key)));
    }
    httpPost.setEntity(new UrlEncodedFormEntity(paramList, Consts.UTF_8));
    httpPost.setConfig(RequestConfig.custom().setConnectTimeout(30000).setSocketTimeout(30000).build());
    response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    int statusCode = response.getStatusLine().getStatusCode();
    if (HttpStatus.SC_OK == statusCode)//如果响应码是 200
        System.out.println(EntityUtils.toString(entity));
} finally {
    if (response != null) {
        response.close();
    }
    if (httpPost != null) {
        httpPost.releaseConnection();
    }
    httpClient.close();
}
}

//创建链接
public static CloseableHttpClient createSSLClientDefault() throws Exception {
    SSLContext sslContext = new SSLContextBuilder().loadTrustMaterial(null, new AllTrustStrategy()).build();
    SSLConnectionSocketFactory sslSf = new SSLConnectionSocketFactory(sslContext);
    return HttpClients.custom().setSSLSocketFactory(sslSf).build();
}

//加载证书
private static class AllTrustStrategy implements TrustStrategy {
```

```
public boolean isTrusted(X509Certificate[] x509Certificates, String s) throws CertificateException {  
    return true;  
}  
}  
}
```

## 8.7 错误码

错误码	描述	备注
-309	校验商户调用密码出错	
-301	校验商户 MAC 校验域出错	
1002	商户未配置密钥信息	
BE100002	受理机构代码为空	
BE110028	分账信息有误	
BE110062	没有找到符合条件的记录。	原订单不存在
BE199999	请求参数有误	具体的参数错误，在描述中有体现
BE300000	商户不存在	
BE300006	受理机构未配置此交易权限	原支付机构不支持退款
BE300007	商户未配置此交易权限	商户没有给定渠道的普通退款权限
BE300012	分账金额总和不等于订单总金额	
BE300013	商户 IP 验证异常	
BE300018	订单不支持部分扣款	原订单支付机构不支持部分退款
BE301001	订单金额出现异常，交易失败	退款金额不符合要求
BE301007	不支持批量支付订单的退款，退款失败	
BE301008	订单的业务类型不支持退款，退款失败	
BE301009	订单状态不为成功，退款失败	
BE301010	退款订单受理中，退款失败	
BE301011	订单已退款或冲正，退款失败	
BE999999	系统繁忙，请稍后再试	系统原因，联系技术支协查
BE110078	原交易不存在或超过退款限制时间	

## 9 退款回调通知接口

### 1. 通知接口说明

通知接口由商户提供;

退款订单状态回调地址以《6.退款接口》中请求参数 (bgUrl) 为准。

### 2. 参数

#### HTTP 表单提交

字段	名称	类型 长度	必填	说明
refundReqNo	退款流水号	30	M	该流水在商户处必须唯一
ourTransNo	翼支付网关 平台交易流水号	8	M	由翼支付网关平台提供, 商户必须保存该信息, 格式: yyyyMMDD, 商户对账、清算报表以此日期为准
oldOrderNo	原扣款订单号	30	M	原扣款成功的订单号
oldOrderReqNo	原订单请求 支付流水号	30	M	原扣款成功的请求支付流水号
merchantCode	商户代码	20	M	由翼支付网关平台统一分配
transAmt	退款交易金额	12	M	单位为分, 小于等于原订单金额
transStatus	退款状态	10	M	退款状态 B 代表退款成功, C 代表退款失败



mac	数字签名	256	M	数字签名算法由翼支付网关平台统一提供，作为核查依据（为 1 时有效）
-----	------	-----	---	------------------------------------

### 3. Sign 验签域说明

sign 校验域说明 MD5: 顺序（注意是字段名是大写）

MERCHANTCODE=4401000041&OLDORDERNO=16091310313809288961&OLDORDERREQNO=16091310313809288961000001&OURTRANSNO=2016091300000667874366&REFUNDREQNO=16091310313809288961000002&TRANSAMT=1&TRANSSTATUS=B&KEY=67B6A6362B7C80E9

### 4. 异步通知说明（重要）

商户平台在收到后台退款结果后，请直接在应答退款通知需商户响应：2016091300000668258065 即 ourTransNo。



## 10 撤销交易接口

### 10.1 请求说明

#### 1. 请求地址

环境	HTTPS请求地址
正式环境	<a href="https://webpaywg.bestpay.com.cn/reverse/reverse">https://webpaywg.bestpay.com.cn/reverse/reverse</a>

#### 2. 请求参数

HTTP 表单提交				
字段	名称	类型长度	必填	说明
merchantId	商户代码	20	M	由翼支付网关平台统一分配给各接入商户
subMerchantId	商户子代码	20	O	商户子代码，如没有则填空
merchantPwd	商户调用密码	20	M	商户执行时需填入相应密码
oldOrderNo	原扣款订单号	30	M	原扣款成功的订单号
oldOrderReqNo	原订单请求支付流水号	30	M	原扣款成功的请求支付流水号
refundReqNo	退款流水号	30	M	该流水在商户处 <b>必须唯一</b>

refundReqDate	退款请求日期	14	M	yyyyMMDD
transAmt	退款交易金额	12	M	单位为分，必须等于原订单金额
channel	渠道	5	M	默认填：05
mac	Mac校验域	32	M	md5摘要，MD5加密获得32位大写字符

3.mac 说明

MAC 校验域说明：MD5: 顺序（注意是字段名是大写） MERCHANTID=123456789&MERCHANTPWD=123456&OLDORDERNO=123456& OLDORDERREQNO=1234567&REFUNDREQNO=200603141210&REFUNDREQDATE=20150720 & TRANSAMT=1&KEY=344C4FB521F5A52EA28FB7FC79AEA889478D4343E4548C02
---

10.2 应答说明

接口响应（      格式）				
字段	名称	类型长度	必填	说明
success	是否成功	5	M	true：成功，false：失败
result	调用返回值		O	结果：{ sign=, refundReqNo=, oldOrderNo=, transAmt=}; sign是签名串；refundReqNo退款请求流水号，oldOrderNo原订单号，transAmt交易金额，分为单位。

errorCode	错误码	10	O	当success为false时取此值，result为空
errorMsg	错误描述	10	O	当success为false时取此值，result为空

## 10.3 响应示例

请求示例

```
oldOrderReqNo=20150713093607000001&oldOrderNo=20150713093607&mac=880D9E9DE657E3C351697D3B6CFE5626&merchantId=043101180050000&transAmt=1&channel=01&refundReqNo=r20150713093607000001&refundReqDate=20150708&merchantPwd=123456
```

响应示例

成功

```
{"success":true,"result":{"sign":"A5FD5994035D2A69F68C1951485D4F6C","refundReqNo":"201611180000002","oldOrderNo":"20161118151028abcd","transAmt":"1"},"errorCode":null,"errorMsg":null}
```

失败

```
{"success":false,"result":null,"errorCode":"BE110062","errorMsg":"没有找到符合条件的记录。"}
```

## 10.4 接口说明

1. 商户要保证冲正流水号的唯一性。
2. 冲正与退款接口不同在于，冲正接口对于除成功状态外的交易，可以使交易作废。当交易已经成功时，冲正接口会申请退款。冲正接口的调用时间仅限于原交易发生的当天，退款不受此限制。
3. MAC 校验，保证了交易信息到翼支付网关平台的完整性，参与 MAC 运算的字符及其顺序如下：

- a. MERCHANTID=123456789&MERCHANTPWD=123456&OLDORDERNO=123456&OLDORDERREQNO=1234567&REFUNDREQNO=200603141210&REFUNDREQDATE=20150720&TRANSA  
MT=1&KEY=123456
  - b. 商户业务平台将上述固定顺序组织的字符串，参数名均为大写，使用标准 md5 算法进行摘要，摘要结果 16 进制字符串，字符串建议使用大写，就是校验域 MAC 的值，将 MAC 值和订单信息调用接口时，一起提供给翼支付网关平台。
4. 对返回的 [10.2](#) 中 result 验签顺序：
- OLDORDERNO=20170119114534659495&REFUNDREQNO=20170119144446990000&TRANSA  
MT=1&KEY=9A5357588EFB80D378403FF3041DCCCCC582C050E527D6

## 10.5 客户端测试用例

```
import com.bestpay.paycenter.commons.util.CryptTool;
import org.apache.http.Consts;
import org.apache.http.HttpEntity;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.config.RequestConfig;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class TestClient {
    public static void main(String[] args) throws Exception {
        String merchantId = "043101180050000";
```

```
String merchantPwd = "111";
String oldOrderNo = "1533734609531";
String oldOrderReqNo = "15337346095231";
String refundReqNo = "20150608113649";
String refundReqDate = "20150720";
String transAmt = "1";
String channel = "02";
```

```
StringBuilder sb = new StringBuilder();//组装 mac 加密明文串
sb.append("MERCHANTID=").append(merchantId);
sb.append("&MERCHANTPWD=").append(merchantPwd);
sb.append("&OLDORDERNO=").append(oldOrderNo);
sb.append("&OLDORDERREQNO=").append(oldOrderReqNo);
sb.append("&REFUNDREQNO=").append(refundReqNo);
sb.append("&REFUNDREQDATE=").append(refundReqDate);
sb.append("&TRANSAMT=").append(transAmt);
sb.append("&KEY=").append("111");//此处是商户的 key
```

String mac = CryptTool.md5Digest(sb.toString());//进行 md5 加密(商户自己封装 MD5 加密工具类，此处只提供参考)

```
Map<String, String> param = new HashMap<String, String>();//组装请求参数
```

```
param.put("merchantId", merchantId);
param.put("merchantPwd", merchantPwd);
param.put("oldOrderNo", oldOrderNo);
param.put("oldOrderReqNo", oldOrderReqNo);
param.put("refundReqNo", refundReqNo);
param.put("refundReqDate", refundReqDate);
param.put("transAmt", transAmt);
param.put("channel", channel);
param.put("mac", mac);
```

```
CloseableHttpClient httpClient = HttpClients.createDefault();
HttpPost httpPost = null;
CloseableHttpResponse response = null;
try {
    httpPost = new HttpPost("https://webpaywg.bestpay.com.cn/reverse/reverse");
    List<NameValuePair> paramList = new ArrayList<NameValuePair> ();
    for (String key : param.keySet()) {
        paramList.add(new BasicNameValuePair(key, param.get(key)));
    }
}
```

```
}
httpPost.setEntity(new UrlEncodedFormEntity(paramList, Consts.UTF_8));
httpPost.setConfig(RequestConfig.custom().setConnectTimeout(30000).setSocketTimeout(30000).build());
response = httpClient.execute(httpPost);
HttpEntity entity = response.getEntity();
int statusCode = response.getStatusLine().getStatusCode();
if (HttpStatus.SC_OK == statusCode)//如果响应码是 200
    System.out.println(EntityUtils.toString(entity));
} finally {
    if (response != null) {
        response.close();
    }
    if (httpPost != null) {
        httpPost.releaseConnection();
    }
    httpClient.close();
}
}
```

## 10.6 错误码

错误码	描述	备注
-309	校验商户调用密码出错	
-301	校验商户 MAC 校验域出错	
1002	商户未配置密钥信息	
BE100002	受理机构代码为空	
BE110028	分账信息有误	
BE110062	没有找到符合条件的记录。	原订单不存在
BE199999	请求参数有误	具体的参数错误，在描述中有体现
BE300000	商户不存在	
BE300006	受理机构未配置此交易权限	原支付机构不支持退款
BE300007	商户未配置此交易权限	商户没有给定渠道的普通退款权限
BE300012	分账金额总和不等于订单总金额	
BE300013	商户 IP 验证异常	
BE300018	订单不支持部分扣款	原订单支付机构不支持部分退款
BE301001	订单金额出现异常，交易失败	退款金额不符合要求

BE301007	不支持批量支付订单的退款，退款失败	
BE301008	订单的业务类型不支持退款，退款失败	
BE301009	订单状态不为成功，退款失败	
BE301010	退款订单受理中，退款失败	
BE301011	订单已退款或冲正，退款失败	
BE301012	订单不为未退款，冲正失败	
BE301013	订单状态为作废，冲正失败	
BE301014	原订单不为当天订单，冲正失败	
BE999999	系统繁忙，请稍后再试	系统原因，联系技术支协查
BE110078	原交易不存在或超过退款限制时间	

## 11 附录说明

### 11.1 交易支付-分账说明

商户若需分账，则填写 ledgerDetail /DIVDETAILS 字段，例如：

a) 规则：023101111:4|023102222:3|023103333:1

b) 说明：订单总金额 4+3+1=8 分,商户 023101111 分账 4 分，每组对应关系之间“|”分隔

c) 分账权限与一般支付权限不同。商户分账支付的权限需要额外申请。

翼支付网关平台分账规则定义如下：

1) 分账支付规则：分账商户必须是分账支付商户的子商户、分账金额必须大于 0 分，最小

分账单位为 1 分、单笔参与分账商户个数不能超过 10 个、单笔交易参与分账商户只能出现一次。

2) 分账退款规则：单商户分账金额必须大于 0 分、退款的分账商户必须在支付分账商户明细中出现，单商户分账金额不大于原商户支付分账金额、单笔交易参与分账商户只能出现一次。

商户 1: 0018888888 父商户

商户 2: 1118888888 子商户

商户 3: 2228888888 子商户

订单金额为 100 分

分账支付正确示例: 0018888888:10|1118888888:80|2228888888|10

分账支付错误示例: 0018888888:0|1118888888:80|2228888888|20

分账支付错误示例: 0018888888:20|1118888888:80|2228888888|0

分账支付错误示例: 0018888888:20|1118888888:70|1118888888:10

以上面正确分账支付为例进行退款 80 分

分账退款正确示例: 0018888888:10|1118888888:60|2228888888|10

分账退款正确示例: 1118888888:70|2228888888|10

分账退款正确示例: 1118888888:80

分账退款错误示例: 0018888888:0|1118888888:80|2228888888|0

分账退款错误示例: 0018888888:0|1118888888:80

分账退款错误示例: 0018888888:0|1118888888:70|2228888888|10

分账退款错误示例: 0018888888:80

分账退款错误示例: 1118888888:70|1118888888:10

多个分账子商户时需加上父商户: 04400000001:0|3401000056:1|1501000004:2

只分给一个子商户时可以省略，分给多个子商户不行。

## 11.2 通知接口（后台）中有 BANKID 字段解释

类别	bankID	解释	备注
个账余额	COMPANYACC_3AC	立减优惠	
个账余额	EPAYTRAVELACC_3AC	翼游账户	
个账余额	EPAYACC	翼支付账户	
个账余额	VOUCHER_3AC	营销代金券	
个账余额	BESTCARDOLD	老翼支付卡	



个账余额	BESTCARD	新翼支付卡	
个账余额	EPAYTRAVELCARD_PRE	翼游卡	
个账余额	EPAYACCWM	翼支付无密账户	
快捷支付	%_RPB2C、%_QB2C、%_Q	快捷支付类型	<p>%为通配符，示例：CCB_D_RPB2C 为建行代扣[借]，其中 CCB 为建设银行缩写；中间的 D 代表着[借]为借记卡，如果为贷记卡，中间没有 D，后面会继续举例；紧接其后的 RP 表示为代扣支付。</p> <p>CMB_RPB2C 为招行代扣[贷]，其中 CMB 为招商缩写；中间没有 D，表示为贷记卡；紧接其后的 RP 表示为代扣支付。</p> <p>CCB_D_QB2C 为建设银行[快捷]，其中 CCB 为建设银行缩写；中间的 D 代表着[借]为借记卡，如果为贷记卡，中间没有 D，后面会继续举例；紧接其后的 Q 表示为快捷支付。</p> <p>CMB_QB2C 为招商银行快捷[信用卡]，其中 CMB 为招商缩写；中间没有 D，表示为贷记卡；紧接其后的 Q 表示为快捷支付。</p> <p>(可以通过是否含有字符串"_D_"来区分是否是借记卡或者信用卡，不能单独通过一个"D"来判断)</p> <p>特例：ECITIC_D、SDB_D、ECITIC、QPCSHB 这几家快捷银行的 BankID 并没有包含_Q，也要并入快捷类银行</p>
个人网银	%_B2C、%、%_C、%_D	对私业务类型	<p>%为通配符，示例：ICBC_B2C_D 为工商银行[借]，其中 ICBC 为中国工商银行缩写；紧急其后的 B2C 表示为对私业务；最后的 D 表示为借记卡支付。</p> <p>BOC_B2C 为中国银行[借贷]，其中 BOC 为中国银行缩写；紧急其后的 B2C 表示为对私业务；最后没有 D 表示为借贷合一的通道，无法区分用户支付的时候是用的借记卡还是贷记卡。</p> <p>ICBC_B2C_C 为工商银行[贷]，其中 ICBC 为中国工商银行缩写；紧急其后的 B2C 表示为对私业务；最后的 C 表示为贷记卡支付。</p> <p>ABC 为中国农业银行[借贷]，其中 ABC 为中国农业银行缩写，不含_C 也不含_D，表示这是借贷合一的通道，无法区分用户支付的时候是用的借记卡还是贷记卡。</p> <p>CMBC_D 为中国民生银行[借]，其中 CMBC 为民生银行缩写，后面含_D 表示这仅支持借记卡的通道，同理如果是_C 表示这仅支持贷记卡的通道</p>
企业网银	%_B2B、%B2B	对公业务类型	<p>%为通配符，示例：ICBC_B2B 为工商银行[企业]，其中 ICBC 为中国工商银行缩写；紧急其后的 B2B 表示为对公业务；对公业务没有[借][贷]之分</p> <p>CMBCB2B 为民生银行[企业]，其中 CMBC 为民生银行缩写，后面含有 B2B 表示这个是对公的企业通道。</p>