

# APACPAY 支付网关 API

## 开发者指南

2016 年 3 月

使用本指南：

- 如何将 APACPAY 对接到你的网站
- 创建和测试 HTML 重定向命令
- 查找国家 ISO 代码，支付错误代码

# 目录

1	引言 .....	3
1.1	为何使用重定向.....	3
1.2	本指南使用对象.....	3
1.3	技能和知识.....	3
1.4	专业术语 .....	3
1.5	注意事项 .....	4
1.6	更多支持 .....	4
1.7	法律声明 .....	4
2	概述 .....	4
2.1	HTML 重定向工作原理.....	5
2.2	商户接入流程 .....	6
3	支付服务 .....	6
3.1	支付接口 .....	6
3.2	发送数据 .....	6
3.3	返回数据 .....	9
4	退款接口 .....	10
4.1	提交参数 .....	10
4.2	返回参数 .....	11
5	查询接口 .....	13
5.1	提交参数 .....	13
5.2	返回参数 .....	13
6	常见错误 .....	15
6.1	I0013: 加密值 (signInfo) 错误.....	15
6.2	I0014: 测试子账户接入正式支付网关 .....	15
6.3	I0015: 正式子账户接入测试支付网关 .....	15
7	附件 .....	16
7.1	货物信息 .....	16
7.2	交易币种 .....	16
7.3	国家代码 .....	17
7.4	测试卡码 .....	17
7.5	错误代码 .....	17
7.6	加密方式 .....	19
7.7	提交示例 .....	20
7.8	版本更新 .....	22
7.9	联系我们 .....	22

# 1 引言

本文档专为APACPAY PTE.LTD.(以下简称APACPAY)支付平台的合作商户提供。目的是帮助商户的技术人员开发相关的接入程序。

本指南介绍了如何在您的网站上对接APACPAY的支付网关，且可支持多平台。

本指南主要体现以下内容：

- 1) 订单如何以HTML的形式发送。
- 2) 顾客如何从商户的网站重定向到APACPAY的支付页面。
- 3) 如何测试商户和APACPAY的对接环境。
- 4) 如何确保您的对接是安全的。

## 1.1 为何使用重定向

使用HTML重定向可以确保：

- 1) 对接APACPAY的支付网关更快，更简单，更便宜。
- 2) 顾客通过APACPAY的安全支付页面支付，而非贵公司支付页面。

## 1.2 本指南使用对象

本技术指南适用对象：

- 1) 管理自己系统对接的商户。
- 2) 其他的技术角色，包括参与设计系统对接的相关人员。

## 1.3 技能和知识

使用HTML重定向与APACPAY对接，所需掌握的基本知识：

- 1) 掌握APACPAY支付服务的基本知识。
- 2) 掌握HTML应用集成相关知识。
- 3) 掌握网页开发的专业知识(包括服务器端脚本)。

## 1.4 专业术语

名称	说明
商户号	签约后，为商家分配的唯一标识号
子账户	商户号下的子商户号
密钥	参数名【signKey】，与子账户相匹配的密钥。
实时返回地址	参数名【returnUrl】，所指定的互联网HTTP地址，当交易完成之后，APACPAY通过重

	定向发送交易结果到该页面去。
异步返回地址	参数名【notifyUrl】，所指定的互联网HTTP地址，当交易完成之后，APACPAY通过重定向异步发送交易结果到该通知地址。将在1天内最多返回5次，每次返回的间隔逐渐增大，当接受到对方返回“OK”时即不再返回。

## 1.5 注意事项

- 1) 连接本系统测试接口，则所有的交易都不会产生费用，不能划款结算。
- 2) 测试订单提交地址和正式订单提交地址请联系技术支持获取。
- 3) 请商户按照 3.2 表中的字段名称提交字段，参数名称区分大小写。
- 4) 发送本系统网关的指令均以 POST 方式发送，并且使用 UTF8 编码格式。
- 5) 接收 APACPAY 支付系统的指令必须能支持 GET 或 POST 方式。

## 1.6 更多支持

你可以通过以下方式了解更多APACPAY产品和服务:

- 1) 访问 APACPAY 的官网 <https://www.apacpay.com>。
- 2) 联系你的客服经理。

## 1.7 法律声明

本文档的解释权归属于APACPAY。若本文档内容变动，恕不另行通知。未得到APACPAY的书面许可，不得以任何目的、任何形式或手段（包括但不限于机械的或电子的）复制或传播本文档的任何部分。对于本文档涉及的技术和产品，APACPAY拥有其专利（或正在申请专利）、商标、版权或其他知识产权，除非得到APACPAY的书面许可协议，本文档不授予这些专利、商标、版权或其他知识产权的许可。

本文档因产品功能示例和描述的需要，所受用的任何人名、企业名和数据都是虚构的，并仅限于APACPAY内部测试使用，不等于APACPAY有对任何第三方的承诺和宣传。

# 2 概述

此节将告诉商户 HTML 重定向是什么，以及如何使用重定向，在商户的系统上接入 APACPAY。HTML 重定向适用于网上购物环境、呼叫中心、预订中心、多渠道的销售情况。它使您能够实时处理付款，并确保您有访问最大数量的使用网上支付。HTML 重定向服务是安全的，并提供给我们可用来评估风险的相关信息。此外，它是最快的方式来获得和运行的在线支付。付款发生在我们的网页上，这样贵公司就不会有昂贵的负担，来收集顾客的数据。你只要把顾客送去我们的支付页面来完成购买。

HTML 重定向是一种简单、安全的集成方法，几乎所有的支付系统都会使用它。



## 2.1 HTML 重定向工作原理

HTML 重定向服务使商户的网上购物系统与 APACPAY 的支付服务共同工作。本指南帮助您创建自己的在线商店。

HTML 重定向使用 HTML 的格式把信息传输到 APACPAY 的支付网关来处理支付。商户在自己的网站上创建表单，并插入订单的相关信息，例如：币种、金额。

以下是 HTML 重定向的工作流程：

- 当购物者准备支付所购买的商品时，商户的网站应该提供一个按钮或链接，当执行时，向我们的付款页面提交订单明细。
- 然后进入 APACPAY 提供的支付页面输入付款详细信息，如信用卡/借记卡的详细信息。购物者的卡信息将通过 APACPAY 提供的支付页面来进行采集。
- 在购物者选择通过信用卡或借记卡支付时
  - APACPAY 将购物者卡相关信息转发给银行进行验证。
  - 银行将返回授权或拒绝响应给 APACPAY。
 如果付款被拒绝，将会给购物者两个选择：尝试另一种付款方式或取消购买。
- 然后将会在支付页面显示交易的结果给购物者，当然我们也可以向他们发送一封确认邮件，这取决于你们的决定。
- APACPAY 也允许将交易的结果以邮件的形式通知商户，这取决于你在 APACPAY 商户接口是如何配置。欲了解更多关于此配置的信息，请联系你的客服经理。

## 2.2 商户接入流程

根据APACPAY提供的商户接入文档参考代码实现您的电子商务系统与APACPAY的电子支付平台实现正确的对接是使用APACPAY支付服务的前提。在接入过程中注意以下接入流程将使您的接入工作达到事半功倍的效果。

- 1) 双方公司洽谈，签订合同。
- 2) 系统建档，分配商户号，子账户，signKey 值给对方。
- 3) 取得开发文档。
- 4) 查看接口开发文档：刚开通的商户号为测试状态，测试时请使用测试订单提交地址。
- 5) 进行接口调试，有任何问题根据本文档不能解决的，请联系技术支持。
- 6) 测试通过后，请联系客服或技术支持，检查通过后，会把商户号转到正式系统。
- 7) 更换正式订单提交地址。
- 8) 测试并上线。

## 3 支付服务

### 3.1 支付接口

支付步骤：

- 1) 商户发起交易请求，使用 http 协议，通过浏览器重定向到支付系统。
  - 2) 支付系统处理交易请求，并提交到银行进行消费交易。
  - 3) 交易完成后，支付系统把支付结果相关参数重定向到商户的交易返回地址。
- 具体的支付代码示例参考 6.7 提交示例。

### 3.2 发送数据

参数名	类型	长度	是否必填	描述
<b>订单信息</b>				
merNo	String	5	Yes	<b>【商户号】</b>
gatewayNo	String	8	Yes	<b>【子账户】</b>
orderNo	String	50	Yes	<b>【商户订单号】</b>
orderAmount	String	10	Yes	<b>【交易总金额】</b> 包括订单的其他所有金额之和,只能为数字。 小数位数请参考7.2交易币种
orderCurrency	String	3	Yes	<b>【交易币种】</b> 币种简称,具体详见附录: 6.2 交易币种 (ISO 4217)
signInfo	String	64	Yes	<b>【签名数据】</b> 明文加密结构: merNo + gatewayNo + orderNo +

				orderCurrency+ orderAmount + returnUrl+ signkey; 返回数据为大写,各语言加密方式不同.具体详见附录: 6.6 sha256加密方式
returnUrl	String	500	Yes	【实时返回地址】返回交易实时信息到商户网店。
notifyUrl	String	500	No	【异步返回地址】返回交易异步信息到商户网店。
shopperId	String	64	Yes	【购物者的ID】
verifyBillingAddress	String	10	No	【是否验证账单信息】YES: 需要验证 NO: 不需要验证, 默认是YES(需要验证)。
platform	String	50	No	【平台】
shopperLevel	String	10	No	【购物者级别】
isOpen3D	String	10	No	【是否验证3D】do3DS: 需要验证3D; no3DS: 不需要验证3D。
token	String	10	No	【一键支付】YES: 需要一键支付; NO: 不需要一键支付。
cardType	String	10	No	【指定卡种】 1:Visa 2:Master 3:JCB 4:AE 5:Diners 6:Maestro
defaultLang	String	10	No	【默认语言】没有指定的话, 默认用浏览器的语言 en_US --英语 zh_HK --中国 香港 zh_CN --中国 de_DE --德国 ja_JP -- 日本 es_ES -- 西班牙 ko_KR -- 韩国 fr_FR -- 法国 it_IT -- 意大利 ru_RU -- 俄罗斯 en_ID -- 印度尼西亚 vi_VN -- 越南 th_TH -- 泰国 ms -- 马来西亚 ar_AE -- 阿拉伯 sv_SE -- 瑞典 tr_TR -- 土耳其 nl_NL -- 荷兰 pt_PT -- 葡萄牙 el_GR -- 希腊 pl_PL -- 波兰

goodsInfo	String	5000	No	【货物详细信息】 包括货物名称，ID，单价，数量
remark	String	1000	No	【备注】 订单备注信息.
<b>账单信息</b>				
firstName	String	100	No	【账单者的名】
lastName	String	50	No	【账单者的姓】
address	String	500	No	【账单者的账单地址】
city	String	100	No	【账单者的账单城市】 全称.
state	String	100	No	【账单者的所在州】 全称.
country	String	100	No	【账单者的账单国家】 简称. 如美国：US
zip	String	100	No	【账单者的邮编】
email	String	200	No	【账单者的邮件】
phone	String	50	No	【账单者的联系 电话】
<b>收货信息</b>				
shipFirstName	String	100	No	【收货人的名】
shipLastName	String	50	No	【收货人的姓】
shipAddress	String	500	No	【收货人的账单地址】
shipCity	String	100	No	【收货人的账单城市】 全称.
shipState	String	100	No	【收货人的所在州】 全称.
shipCountry	String	100	No	【收货人的账单国家】 简称. 如美国：US
shipZip	String	100	No	【收货人的邮编】
shipEmail	String	200	No	【收货人的邮件】
shipPhone	String	50	No	【收货人的联系 电话】



### 3.3 返回数据

支付完成后，系统把支付结果重定向到返回地址(包括 POST 和 GET 两种方式返回)，所以返回页面必须确保两种支付返回方式都能接收。其中：默认先浏览器返回，后服务器 POST 返回。

参数名称	类型	描述
merNo	String	【商户号】
gatewayNo	String	【子账户】
tradeNo	String	【交易流水订单号】
orderNo	String	【商户订单号】
orderCurrency	String	【交易币种】
orderAmount	String	【交易金额】
cardNumber	String	【交易卡信息】
secureVerifyResult	String	<b>【3D验证结果】</b> --Authentication Unavailable --Failed --Authentication offered but not used --Unable to Complete Authentication
orderStatus	String	<b>【交易状态】</b> -2: 待确认 -1: 待处理 0: 失败 1: 成功 4000: 拒付请求 4010: 拒付申诉失败 4020: 拒付申诉成功 4100: 退款请求 4110: 退款成功 4120: 退款失败
orderInfo	String	【交易结果信息】 Code+具体信息。
billAddress	String	<b>【账单地址】</b> 支付成功返回交易的账单地址。 支付失败则返回空值。
signInfo	String	<b>【数据签名】</b> signInfo=sha256(merNo + gatewayNo + tradeNo + orderNo + orderCurrency + orderAmount + orderStatus + orderInfo + signkey);
paymentMethod	String	支付方式
returnType	String	1: 浏览器实时返回。 2: 服务器实时返回。 3: 服务器异步返回。
Remark	String	<b>【备注】</b> 发送数据的时候，传输的什么，就返回什么。

## 4 退款接口

你可以使用APACPAY提供的退款接口进行批量（单次最多提交50笔）退款，另外商户也可以在APACPAY的商户后台进行人工申请退款。

### 4.1 提交参数

参数名称	类型	长度	描述
merNo	String	5	【商户号】
gatewayNo	String	8	【子账户】
refundOrders	List<RefundBean>	50	【退款订单集合】最大数量为 50
RefundBean.tradeNo	String	20	【交易订单号】
RefundBean.orderNo	String	20	【商户订单号】
RefundBean.currency	String	5	【退款币种】
RefundBean.tradeAmount	String	10	【交易金额】
RefundBean.refundAmount	String	10	【退款金额】
RefundBean.refundReason	String	500	【退款原因】
signInfo	String	64	<b>【数字签名信息】</b> 明文加密结构: (merNo + gatewayNo + tradeNo1 + tradeNo2 ...+ signkey); 在进行 SHA256 加密

退款接口请求示例：

```
{
  "merNo":"10005",
  "gatewayNo":"10005001",
  "signInfo":"5648b4de35d4783bda75d0ff574aee007198c5673b55a8e2600ed6267a1c8f53"
  "refundOrders":[
    {
      "tradeNo":"1941996290307513125",
      "orderNo":"1462430729923",
      "currency":"USD",
      "tradeAmount":"2.00",
      "refundAmount":"1.00",
      "refundReason":"test"
    },
    {
      "tradeNo":"1941996290307513126",
      "orderNo":"1462430729924",
      "currency":"USD",
      "tradeAmount":"4.00",
      "refundAmount":"4.00",
    }
  ]
}
```

```

    "refundReason": "test1"
  }
  .....
]
}

```

## 4.2 返回参数

查询接口返回的是 json 格式：

参数名称	类型	描述
merNo	String	【商户号】
gatewayNo	String	【子账户】
errorStatus	String	【错误状态】 返回数字：0: 失败 1: 成功
errorInfo	String	【错误信息】
refundOrders	List<RefundBean>	【退款订单集合】
RefundBean.tradeNo	String	【交易订单号】
RefundBean.orderNo	String	【商户订单号】
RefundBean.currency	String	【订单退款币种】
RefundBean.tradeAmount	String	【订单交易金额】
RefundBean.refundAmount	String	【订单退款金额】
RefundBean.refundReason	String	【订单退款原因】
RefundBean.refundStatus	String	【订单退款返回状态】 返回数字：0: 失败 1: 成功
RefundBean.refundInfo	String	【订单退款返回信息】
signInfo	String	【数字签名信息】 明文加密结构： (merNo + gatewayNo + tradeNo1 + tradeNo2 ...+ signkey); 在进行 SHA256 加密

退款接口返回示例：

```

{
  "merNo": "10005",
  "gatewayNo": "10005001",
  "errorStatus": "1",
  "errorInfo": "OK",
  "refundOrders": [
    {
      "tradeNo": "1941996290307513125",
      "orderNo": "1462430729923",
      "currency": "USD",
      "tradeAmount": "2.00",
      "refundAmount": "1.00",
      "refundInfo": "ok",
      "refundReason": "test ",
      "refundStatus": "1"
    }
  ]
}

```

```
    },
    { "tradeNo": "1941996290307513126",
      "orderNo": "1462430729924",
      "currency ": "USD",
      "tradeAmount": "4.00",
      "refundAmount": "4.00",
      "refundInfo": "申请处理中,不能退款",
      "refundReason": "test1",
      "refundStatus": "0",
    }
    .....
  ],
  "signInfo": "5648b4de35d4783bda75d0ff574ace007198c5673b55a8e2600ed6267a1c8f53"
}
```

## 5 查询接口

俗称勾对，通过查询接口来确认交易状态的复查操作。

当网站的订单状态与支付系统的状态不一致的情况，如出现掉单情况（在APACPAY商户后台查看是已付款状态，而在商户网站中仍是待付款状态），就可以使用该查询接口，对网站异常订单状态进行修正。但是开发查询接口前需要提前把IP提交到我们系统进行申请。

### 5.1 提交参数

参数名称	类型	长度	描述
merNo	String	5	【商户号】
gatewayNo	String	8	【子账户】
orderNo	String	1000	【商户订单号】 网站上的订单号，可以最大查询 100 笔交易。 请使用", " 分隔开。
signInfo	String	64	【数字签名信息】 明文加密结构： (merNo + gatewayNo + signkey); 返回数据为大写.各语言加密方式不同。 详见附录:6.6 sha256 加密方式

### 5.2 返回参数

查询接口返回的是 json 格式：

参数名称	类型	描述
merNo	String	【商户号】
gatewayNo	String	【子账户】
orderNo	String	【商户订单号】
tradeNo	String	【流水订单号】
tradeDate	String	【交易时间】格式： yyyyMMddHHmmss
tradeAmount	String	【交易金额】
tradeCurrency	String	【交易币种】
sourceWebSite	String	【交易网站】

queryResult	String	<b>【交易结果】</b> -2: 待确认 -1: 待处理 0 : 失败 1 : 成功 2 : 订单不存在
returnStatus	String	1 : 成功返回 3 : 传入的参数不完整 4 : 订单查询过多,(最多只能查询 100 笔) 5 : 商户号或者子账户不存在 6 : signInfo 加密信息错误 7 : 商户服务器 IP 未登记 999: 系统异常

查询接口返回示例:

```
{ "merNo":"10000",
  "gatewayNo":"10000001",
  "tradeList":[{"tradeDate":"1941996290307513125",
    "orderNo":"00001",
    "tradeNo":"751312501436375",
    "sourceWebSite":"mytest.com",
    "tradeAmount":"323",
    "queryResult":"1",
    "tradeCurrency":"USD"
  },
  {"tradeDate":"20140312132614",
    "orderNo":"00002",
    "tradeNo":"751312501436376",
    "sourceWebSite":"mytest.com",
    "tradeAmount":"62",
    "queryResult":"0",
    "tradeCurrency":"USD"
  }
  .....
],
  "returnStatus":"1"
}
```

## 6 常见错误

### 6.1 I0013：加密值（signInfo）错误

- 1) 请检查 signKey 正确性，请保证大小写一致。
- 2) 请使用 sha256 加密方法。
- 3) returnUrl 参数里请不要带有 & 符号。

### 6.2 I0014：测试子账户接入正式支付网关

- 1) 商户号开通之后，提供的网接入号是为测试状态。
- 2) 开发接口程序的时候，请使用测试的支付网关。
- 3) 当测试完成之后可以联系客服或者技术支持把商户号状态改为正式。

### 6.3 I0015：正式子账户接入测试支付网关

通知客服把测试账号改为正式账号之后，请同步把接口文件中的测试订单提交地址改为正式订单提交地址。

## 7 附件

### 7.1 货物信息

货物信息(goodsInfo)排列顺序为：货物名称、货物 ID、单价、数量，之间的顺序不能调换。并且单个货物信息之间以“#,#”分开，多个货物之间以“#;#”进行分割。如果部分货物信息或者某个货物信息部分属性为空的话，可用“0”进行代替，此外货物信息中币种与订单币种一致。

需要注意的是：#,#和#;#都为半角状态下字符。

具体实例如下：

```
Good1-Name #,# Good1-ID#,# Good1-Price #,# Good1-Qty
#;#
Good2-Name #,# Good2-ID#,# Good2-Price #,# Good2- Qty
#;#
Good3-Name #,# Good3-ID#,# Good3-Price #,# Good3- Qty
.....
```

### 7.2 交易币种

APACPAY 支付服务接受以下币种，完整的 ISO 4217 币种代码请参考 <http://www.iso.org>

代码	英文描述	中文描述	小数位数
ARS	Nuevo Argentine Peso	阿根廷比索	2
AUD	Australia Dollar	澳元	2
BRL	Brazilian Real	巴西雷亚尔	2
CAD	Canadian Dollar	加元	2
CHF	Swiss Franc	瑞士法郎	2
CNY	Chinese Yen	人民币	2
CZK	Czech Koruna	捷克克朗	2
DKK	Danish Krone	丹麦克朗	2
EUR	Euro	欧元	2
GBP	Pound	英镑	2
HKD	Hong Kong Dollar	港币	2
ILS	New Shekel	以色列币-新谢克尔	2
INR	Indian Rupee	印度卢比	2
IDR	Indonesian Rupiah	印度尼西亚盾	0
JPY	Japanese Yen	日元	0
KRW	Korean Won	韩元	0
MYR	Malaysian Ringgit	马币	2
NOK	Norwegian Krone	挪威克郎	2



NZD	New Zealand Dollar	新西兰元	2
PHP	Philippine Peso	菲律宾比索	2
PLN	Polish Zloty	波兰兹罗提	2
RUB	Russian Ruble	俄罗斯-卢布	2
SEK	Swedish krone	瑞典克朗	2
SGD	Singapore Dollar	新加坡币	2
THB	Thailand Baht	泰铢	2
TRL	Turkish lira	土耳其里拉	2
TWD	Taiwan Dollar	台币	2
UAH	Ukrainian Hryvnia	乌克兰瑞夫尼亚	2
USD	US Dollar	美元	2
VND	Vietnamese New Dong	越南盾	0
ZAR	South African Rand	南非-兰特	2

### 7.3 国家代码

完整的ISO 4217国家代码请参考<http://www.iso.org> APACPAY不负责其内容的维护。

### 7.4 测试卡号

卡类型	卡号
Airplus	122000000000003
American Express	34343434343434
Cartebleue	5555555555554444
Dankort	5019717010103742
Diners	36700102000000 and 36148900647913
Discover card	6011000400000000
JCB	3528000700000000
Laser	630495060000000000 and 630490017740292441
Maestro	6759649826438453 and 67999990100000000019
MasterCard	5555555555554444 and 5454545454545454
Visa	4444333322221111, 49118300000000 and 4917610000000000
Visa Debit	4462030000000000 and 4917610000000000003
Visa Electron (UK only)	4917300800000000
Visa Purchasing	4484070000000000

### 7.5 错误代码

代码	错误信息
I0001	商户号不能为空

I0002	子账户不能为空
I0003	加密值得不为空
I0004	商户号与子账户不匹配
I0005	商户号未激活请联系客服
I0006	商户号停用。请联系客服。
I0007	商户号不存在。
I0008	商户号状态异常
I0009	子账户未激活
I0010	子账户停用
I0011	子账户删除
I0012	子账户状态异常
I0013	加密值错误
I0014	非正式子账户接入正式接口
I0015	非测试子账户接入测试接口
I0016	该子账户未绑定本接口
I0017	商户订单号为空
I0018	订单号不能超过 50 位
I0019	订单金额为空
I0020	订单金额不正确
I0021	订单金额小数位数不能超过 2 位且不能小于 0
I0022	订单币种为空
I0023	订单币种错误
I0024	订单返回地址为空
I0025	订单返回地址长度不能超过 1000 个字符
I0061	商户订单号重复
I0062	流水订单号重复
I0067	注最多只能输入 500 个字符
R0000	高风险
R0001	网站未审核/注册
R0002	跨国支付
R0003	未通过账单校验
C0001	商户子账户未绑定通道
C0002	商户子账户未设置扣率
C0003	商户子账户通道扣率设置错误
C0004	商户子账户通道扣率设置错误
C0005	币种未设置汇率
C0006	获取汇率失败
C0007	商户未绑定支付域名
C0008	商户子账户未绑定 {0} 通道
C0009	商户子账户未设置扣率
S0001	保存到异常交易表失败
S0002	保存到持卡人信息表失败

S0003	保存到非正式交易表失败
S0004	保存到附加交易表失败
S0005	获取通道信息异常
S0006	保存交易信息失败
S0007	更新测试表失败
S0008	删除异常记录表失败
S0009	保存交易记录表失败
S0010	获取商户绑定域名信息失败
S0011	保存持卡人信息表失败
S0012	获取银行通道信息失败
S0013	银行返回修改交易记录表失败
S0014	通道未绑定邮件域名
S0015	更新交易记录表重复支付标识失败
E0001	操作超时
E0002	操作超时
E0003	发送银行失败
E0004	调用银行异常
E0005	银行返回的通道代码不存在
E0006	返回银行失败

## 7.6 加密方式

### 1) Java 加密方式:

```
signInfo=sha256(merNo+gatewayNo+orderNo+orderCurrency+orderAmount+returnUrl+signkey);
```

### 2) PHP 加密方式:

```
$signInfo=hash("sha256",$merNo.$gatewayNo.$orderNo.$orderCurrency.$orderAmount.$returnUrl.$signkey);
```

### 3) C#加密方式:

```
public string GetSHA256(string strData)
{
    //使用SHA256加密算法:
    System.Security.Cryptography.SHA256 sha256 = new System.Security.Cryptography.SHA256Managed();
    byte[] sha256Bytes = System.Text.Encoding.Default.GetBytes(strData);
    byte[] cryString = sha256.ComputeHash(sha256Bytes);
    string sha256Str = string.Empty;
    for (int i=0;i<cryString.Length;i++)
    {
        sha256Str += cryString[i].ToString("X2");
    }
    return sha256Str;
}
```

## 7.7 提交示例

### JAVA 示例:

```
String merNo = request.getParameter("merNo");           // 商户号
String gatewayNo = request.getParameter("gatewayNo");    // 子账户
String signKey = request.getParameter("signKey");        // 密钥
String orderNo = request.getParameter("orderNo");        // 订单编号
String orderAmount = request.getParameter("orderAmount"); // 订单金额
String orderCurrency = request.getParameter("orderCurrency"); // 订单币种
String returnUrl = request.getParameter("returnUrl");     // 商户交易返回地址
//加密方式: SHA256 (商户号 + 子账户 + 订单号 + 订单币种 + 订单金额 + 返回地址 + key)
//加密结果转换大写
String signInfo = SHA256(merNo+gatewayNo+orderNo+orderCurrency+
orderAmount+returnUrl+signKey).toUpperCase();

HttpClient http_client = new HttpClient();
String requestUrl = "";                                // 请求交易地址
PostMethod postMethod = new PostMethod(requestUrl);
postMethod.getParams().setParameter(HttpMethodParams.RETRY_HANDLER,
new DefaultHttpClientRetryHandler());
NameValuePair[] data = {
    new NameValuePair("merNo", merNo),
    new NameValuePair("gatewayNo", gatewayNo),
    new NameValuePair("signKey", signKey),
    new NameValuePair("orderNo", orderNo),
    new NameValuePair("orderAmount", orderAmount),
    new NameValuePair("orderCurrency", orderCurrency),
    new NameValuePair("signInfo", signInfo),
    new NameValuePair("returnUrl", returnUrl)
};
postMethod.setRequestBody(data);
String content = "";
int statusCode = 0;
try {
    postMethod.getParams().setContentCharset("UTF-8");
    statusCode = http_client.executeMethod(postMethod);
    content = postMethod.getResponseBodyAsString();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    postMethod.releaseConnection();
}
```

## PHP 示例

```
$merNo = 'Merchant Number';           //商户号
$gatewayNo = 'Subaccount';             //子账户
$signKey = 'nnl468zx';                 //密钥
$orderNo = '2016012300000001';         //订单编号
$orderAmount = '100';                  //订单金额
$orderCurrency = 'AUD';                 //订单币种
$returnUrl = 'merchant return url';     //商户交易返回地址
//加密方式: SHA256 (商户号 + 子账户 + 订单号 + 订单币种 + 订单金额 + 返回地址 + key)
//加密结果转换大写
$signInfoOld = $merNo.$gatewayNo.$orderNo.$orderCurrency.$orderAmount.$returnUrl.$signKey;
$signInfoNew = strtoupper(hash('sha256', $signInfoOld));

$requestUrl = 'request url';            //请求交易地址
$post_data = array(
    'merNo' => $merNo,
    'gatewayNo' => $gatewayNo,
    'signKey' => $signKey,
    'orderNo' => $orderNo,
    'orderAmount' => $orderAmount,
    'orderCurrency' => $orderCurrency,
    'returnUrl' => $returnUrl,
    'notifyURL' => $notifyURL,
    'signInfo' => $signInfoNew
);
send_post($requestUrl, $post_data);      //PHP 发送 POST 请求
function send_post($url, $post_data) {
    $postdata = http_build_query($post_data);
    $options = array(
        'http' => array(
            'method' => 'POST',
            'header' => 'Content-type:application/x-www-form-urlencoded',
            'content' => $postdata,
            'timeout' => 15 * 60          // 超时间 (单位:s)
        )
    );
    $context = stream_context_create($options);
    $result = file_get_contents($url, false, $context);
    return $result;
}
```

非直连提交示例:

```
<form name="payForm" action="requestUrl" method="post">
<input type="hidden" name=" merNo" value=" merNo" />
<input type="hidden" name=" gatewayNo" value="gatewayNo" />
<input type="hidden" name=" orderNo" value="orderNo" />
<input type="hidden" name=" orderAmount" value="orderAmount" />
<input type="hidden" name=" orderCurrency" value="orderCurrency" />
<input type="hidden" name=" returnUrl" value="returnUrl" />
<input type="hidden" name=" signInfo" value="signInfo" />
</form>
```

## 7.8 版本更新

版本	日期	内容
1.2	Mar 2016	更新: <ul style="list-style-type: none"><li>• 增加国家代码申明.</li></ul>
1.1	Jan 2016	更新: <ul style="list-style-type: none"><li>• 处理异常.</li></ul>
1.0	May 2015	创建: <ul style="list-style-type: none"><li>• 创建中文版开发者指南.</li></ul>

## 7.9 联系我们

要了解更多信息, 请与您的客户经理联系或电子邮件[media@apacpay.com](mailto:media@apacpay.com)

©APACPAY 2017年保留所有权利。

APACPAY商标及任何相关的品牌名称属APACPAY公司所有。