*Funpay payment gateway*

*Funpay支付网关商家接入说明 V1.3*

*FUNPAY PAYMENT GATEWAY MERCHANT INTEGRATION SPECIFICATION V1.3*

修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **时间** | **修改者** | **修改纪要** | **审核者** |
| V1.0 | 2019-1-23 | Lotiis | 1.创建文档 | Lotiis |
| V1.1 | 2019-3-7 | Lotiis | 1. 消除了一些容易给合作伙伴造成误解的地方 2. 修改了新的返回结果协议 3. API版本升级到V1.1 | Lotiis |
| V1.1 | 2019-3-11 | Lotiis | 1.增加了打款的功能 | Lotiis |
| V1.2 | 2019-3-29 | Lotiis | 1.升级了线下还款的功能  2.增加了交易结果查询的功能 | Lotiis |
| V1.3 | 2019-4-3 | Lotiis | 1.增加了必要的LOGO存储路径  3.升级了打款的接口 | Lotiis |
| V1.31 | 2019-5-5 | Lotiis | 1.修改了部分错误信息  2.增加了核验银行信息的接口 | Lotiis |
| V1.32 | 2019-5-8 | Lotiis | 1.修改了打款接口的相关说明  2.新增了虚拟银行卡的接入说明 | Lotiis |
| V1.32 | 2019-6-3 | Lotiis | 1.升级了银行账户核验接口 | Lotiis |
| V1.33 | 2019-6-17 | Lotiis | 1.新增了参数clientID  2.修改了returnUrl说明  3.打款新增设置异步回调的参数  4.更改了Funpay域名 | Lotiis |
| V1.34 | 2019-8-16 | Lotiis | 1. 新增更新线下支付码的功能  2. 修改虚拟卡使用的相关说明  3. 新增虚拟卡交易记录查询功能 | Lotiis |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[*一、* *概述* 3](#_Toc17814422)

[*二、* *账户申请* 3](#_Toc17814423)

[*三、* *API加密规则* 4](#_Toc17814424)

[*四、* *线下支付* 5](#_Toc17814425)

[*五、* *打款功能* 10](#_Toc17814426)

[*六、* *获取银行以及支行列表* 13](#_Toc17814427)

[*七、* *核验银行卡信息功能* 15](#_Toc17814428)

[*八、* *虚拟银行卡功能* 16](#_Toc17814429)

[*九、* *交易结果查询功能* 25](#_Toc17814430)

[*十、* *自动对账功能* 30](#_Toc17814431)

[*十一、* *结果返回码表* 30](#_Toc17814432)

[*十二、* *订单状态码表* 30](#_Toc17814433)

# 概述

1. 文档对象

本文档适用于所有接入Funpay支付网关的合作伙伴的技术人员。

1. 文档目的

本文档主要用于向合作伙伴的技术人员说明Funpay支付网关的接入方法以便合作伙伴能快速便捷的接入。

1. 网关说明

Funpay支付网关是一个多功能的支付解决方案，既解决了用户向商家支付的问题，也解决了商家向用户进行付款的问题。同时为了方便用户使用，Funpay同时提供了线上网银支付和线下网点支付两种付款方式。

1. 术语说明
2. 商家 merchant：即需要接入Funpay支付的所有合作伙伴
3. 用户、终端用户End user：使用商家服务、APP的最终用户
4. 线下支付 offline payment：通过线下POS机、便利店网点等方式完成的支付交易
5. 支付功能：指终端用户对接入商家发起的支付。
6. 打款功能：指接入商家对终端用户发起的支付。
7. 商家服务器 merchant server：商家自有的后台系统，用于接收交易并处理交易结果等
8. 银行系统Bank system：泛指发行并管理用户进行支付的银行卡、信用卡的银行的后台结算系统
9. 便利店POS：泛指线下支付网点和前端线下支付手持设备。
10. 重要说明

若无特殊说明，本文档中传输的字符串均使用“UTF-8”编码。

# *账户申请*

1. 请与我们商务人员联系，我们商务人员会与您接洽，并为您创建账户，然后将账户与初始登录密码返还给您。
2. 请准备好您的现金提现银行账户、业务名称、计费点详情等信息。
3. 请登录我们管理后台，创建您的业务和计费点，并完善您的银行账户信息。请注意如果取款信息未完善，我们将无法为您进行结算。另外如果银行信息完善后，为了您的资金安全，我们的后台将不再为您提供修改的功能。如果有特殊原因确实需要修改，需要联系我们的商务人员。
4. 相关信息设置完成后需要联系我们的商务人员，我们的商务人员需要对您的输入信息进行核对确认，通过后，我们系统将会为您生成相关数据。
5. 再次登录我们后台即可获取您业务和计费点的专有ID以及安全密钥。
6. 请注意此安全密钥非常重要，为了您的财产安全请妥善保管，并不要写在前端，而是使用的时候从您的后台拉取。
7. 在注册业务的时候需要您提供一个回调地址，由于安全需要，建议您使用安全性更高的SSL、TSL证书的HTTPS（不是强制性要求），并在后台设置的时候加上 “https://” 或者 “http://”。

# API加密规则

为了确保交易的安全性，本系统对所有交互API采用了一套加密机制，所有API交互均需要使用此套加密规则，以确保合作伙伴资金财产的安全性。加密规则具体如下：

1. 从Funpay管理后台获取到secret key，保存在服务器合适的问题。
2. 前端从服务器获取到secret key，保存在内存中。
3. 根据API要求，以“key=value”列出所有参数名称与值，形成键值对组，设为Array1。
4. 将Array1按照ASCII码表从小到大的顺序对“key” 进行排序，得到新的键值对，设为Array2。
5. 按照排好的顺序，以连接符号“&”，串联所有的键值对，形成字符串String1。
6. 在String1后面再串入您的secret key，形成String2。
7. 对String2进行摘要运算（目前仅支持MD5运算，后续会加入SHA1、SHA256的支持），获得到新的字符串String3。
8. 将String3作为一个新的值赋给可key ”sign”，形成新的键值对，并将此键值对插入到键值对组Array1，形成新的键值对组Array3。
9. Array3即为需要上传到 Funpay所需要的所有参数。
10. 将Array3生成一个URL参数的字符串，然后取BASE64编码，请注意，这里需要使用URL Safe Base64，新生成字符串String4。
11. 将String4赋值给Key “param”即为请求的参数。

例子：

1. 假设上传的参数表如下：

ccc=111 aaa=22 aab=444 eee=555 ddd=666 🡨即为Array1

1. 将Array1按照key的ASCII码从小到大的顺序排列

aaa=22 aab=444 ccc=111 ddd=666 eee=555 🡨即为Array2

1. 将Array2用“&”符号连接

aaa=22&aab=444&ccc=111&ddd=666&eee=555 🡨即为String1

1. 在String1后面加入secret key，假设为“PslBum8DW0uGEI687X38QpdW53e452zvVVYyzUcT”

aaa=22&aab=444&ccc=111&ddd=666&eee=555PslBum8DW0uGEI687X38QpdW53e452zvVVYyzUcT 🡨即为String2

1. 对String2取MD5（大写字母）

MD5(aaa=22&aab=444&ccc=111&ddd=666&eee=555PslBum8DW0uGEI687X38QpdW53e452zvVVYyzUcT) === 95E3862B4D4D22ABA2DE461A330B3608🡨即为String3

1. 将String3赋值给“sign”并插入Array1得到新的键值对

ccc=111 aaa=22 sign=95E3862B4D4D22ABA2DE461A330B3608 aab=444 eee=555 ddd=666 🡨即为Array3

1. 如果HTTP方法为GET则进行[8）](#步骤八)和[9）](#步骤九),如果HTTP方法为POST则直接看[10）](#步骤十)
2. 将Array3用“&”连接串成字符，并取Base64值

safe\_Base64(ccc=111&aaa=22&sign=95E3862B4D4D22ABA2DE461A330B3608&aab=444&eee=555&ddd=666)= Y2NjPTExMSZhYWE9MjImc2lnbj05NUUzODYyQjRENEQyMkFCQTJERTQ2MUEzMzBCMzYwOCZhYWI9NDQ0JmVlZT01NTUmZGRkPTY2Ng 🡨即为String3

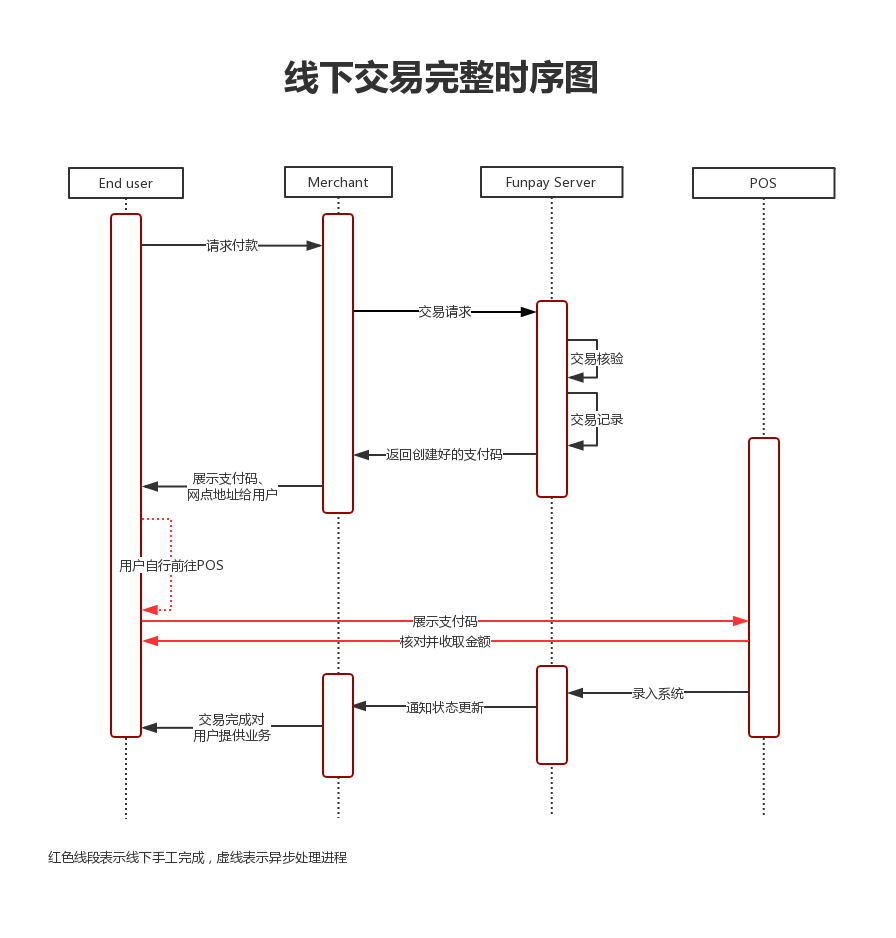
1. String3赋值给“param”作为参数拼接成新的URL

param=Y2NjPTExMSZhYWE9MjImc2lnbj05NUUzODYyQjRENEQyMkFCQTJERTQ2MUEzMzBCMzYwOCZhYWI9NDQ0JmVlZT01NTUmZGRkPTY2Ng Y2NjPTExMSZhYWE9MjImc2lnbj05NUUzODYyQjRENEQyMkFCQTJERTQ2MUEzMzBCMzYwOCZhYWI9NDQ0JmVlZT01NTUmZGRkPTY2Ng

1. 如果HTTP方法为POST则不需要取Base64，直接将相关参数插入JSON数据即可。

# 线下支付

1. 线下支付流程



1. 注意事项

用户使用Viettel Post还款的时候，需要告知店员通过Funpay进行还款，所以在商家页面需要引导用户告知Viettel Post店员使用FunPay支付。如需要FunPay Logo可在以下地址获取 （透明背景,Size1024\*1024）

<https://pic.funpay.asia/funpay.png>

如果需要其他尺寸可以如此获取

<https://pic.funpay.asia/funpay.png?x-oss-process=image/resize,w_1024>

其中1024为您实际需要的宽高，会自动等比缩小。

如果白色背景jpg可以在如下地址获取:

<https://pic.funpay.asia/funpay.jpg>

同样支持自动等比例缩放。

如果您需要Viettel Post 的Logo可以在在这个地址获取:

<https://pic.funpay.asia/vtp.png>

同样支持自动等比例缩放。

1. 线下支付接口
2. API地址地址

[https://payment.funpay.asia/fun/payment/offlinePay](https://payment.funmobi.asia/fun/payment/offlinePay)

1. 使用的方法

GET

1. 所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| feeID | number | Y | 计费点ID，funpay后台分配 |
| clientID | string | N | 用户ID，商户传入，以便区分用户对账平账，商户可以根据自己需要传入终端用户的ID或者合同号等唯一标记，Funpay对此字段不作区分处理，返回结果中亦不进行返回，仅用于结果报表中的检索。商户可以根据自己需求决定是否传入。如果不传入请不要使用此字段。如果带了此字段请务必确保其有值。 | |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| amount | number | Y | 支付金额，单位1VND |
| currency | string | Y | 货币符号，必须为“VND” |
| name | string | Y | 计费点名称，必须与申请计费点的时候输入的一致 |
| orderNo | string | Y | 商家自定义订单号，唯一，最长40字符，仅数字字母，不支持特殊字符 |
| expireDate | string | Y | 商家自行指定的过期时间，格式yyyyMMddHHmmss。超过指定时间后用户将不能再对此订单进行支付。 |
| returnUrl | url | Y | 回调函数地址，立即回调用，不需要URL编码，因为不会直接传输链接地址。 此参数必须为“http://”开头或者“https://”开头，否则不会触发回调。强烈建议就算您不依赖回调，也填入一个合法的回调地址，可以不做任何逻辑处理，但是方便进一步扩展。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |
| purchaseType | number | Y | 选择使用的支付方式，此处使用固定值2。 |
| phoneNumber | string | Y | 用户手机号码，用于线下便利店跟用户确认核验订单，避免误操作 |
| userName | string | Y | 用户真实姓名，用于线下便利店跟用户确认核验订单，避免误操作 |

1. 返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#线下Result数据结构)” |

1. 回调
2. 回调说明

回调会有两次，一次是在支付完成后同步返回，一次是在交易完成后，异步回调 。第一次回调为商户传入的参数returnUrl，第二次回调地址为管理后台页面设置的回调地址。

1. 回调方法

POST

1. 回调参数，两次回调完全相同（使用JSON数据结构组织），

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#线下Result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID，商家传入，透传回来 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| name | string | N | 计费点名称，商家传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| status | int | Y | 订单状态，详见[订单状态码表](#_状态码表)。 |
| code | string | N | 需要展示给用户的交易码，用户可以根据此码去便利店进行支付，交易成功后此码不再有效，因此支付结果回调的时候不一定会再次返回此码。 |
| shopLink | url | N | 便利店地址，商家需要将此链接展示给用户，用户点击后可以用webview打开相关页面，以展示其附近的便利店地图。交易成功后此码不再有效，因此支付结果回调的时候不一定会再次返回此码。 |

1. 更新线下支付码
2. 说明

本接口用于更新已经创建的支付码，使用场景是用户的交易需要延期或者支付金额发生了改变，另外商家回调地址等需要改变也可以通过此接口来触发。

请注意，已经支付或者已经过期的支付码不可以再通过此接口进行更新。

1. 接口地址

[https://payment.funpay.asia/fun/payment/updateVTPCode](https://payment.funmobi.asia/fun/payment/updateVTPCode)

1. 使用的方法

POST

1. 所需参数（以JSON结构体传输）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| feeID | number | Y | 计费点ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| tradeNo | string | Y | funpay返回的交易号 |
| amount | string | N | 如果商家需要更新交易的金额可以使用此参数进行设置，如不需要更新则不传入此参数或者值置空。 |
| expireDate | string | N | 商户自行指定的新的过期时间，格式yyyyMMdd，以河内时间为准。如不需要更新则不传入此参数或者值置空。 |
| returnUrl | string | N | 回调地址，Funpay将以此地址通知用户进来的每一次汇款。此参数必须为“http://”开头或者“https://”开头，否则不会触发回调。如不需要更新则不传入此参数或者值置空。 |
| phoneNumber | string | N | 新的用户手机号。如不需要更新则不传入此参数或者值置空。 |
| userName | string | N | 新的用户姓名。如不需要更新则不传入此参数或者值置空。 |
| status | number | N | 还款码状态设置，-1代表该还款码失效。不设置或者设置为其他值则代表保持原状态不变。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |

1. 返回值（以JSON数据结构组织）

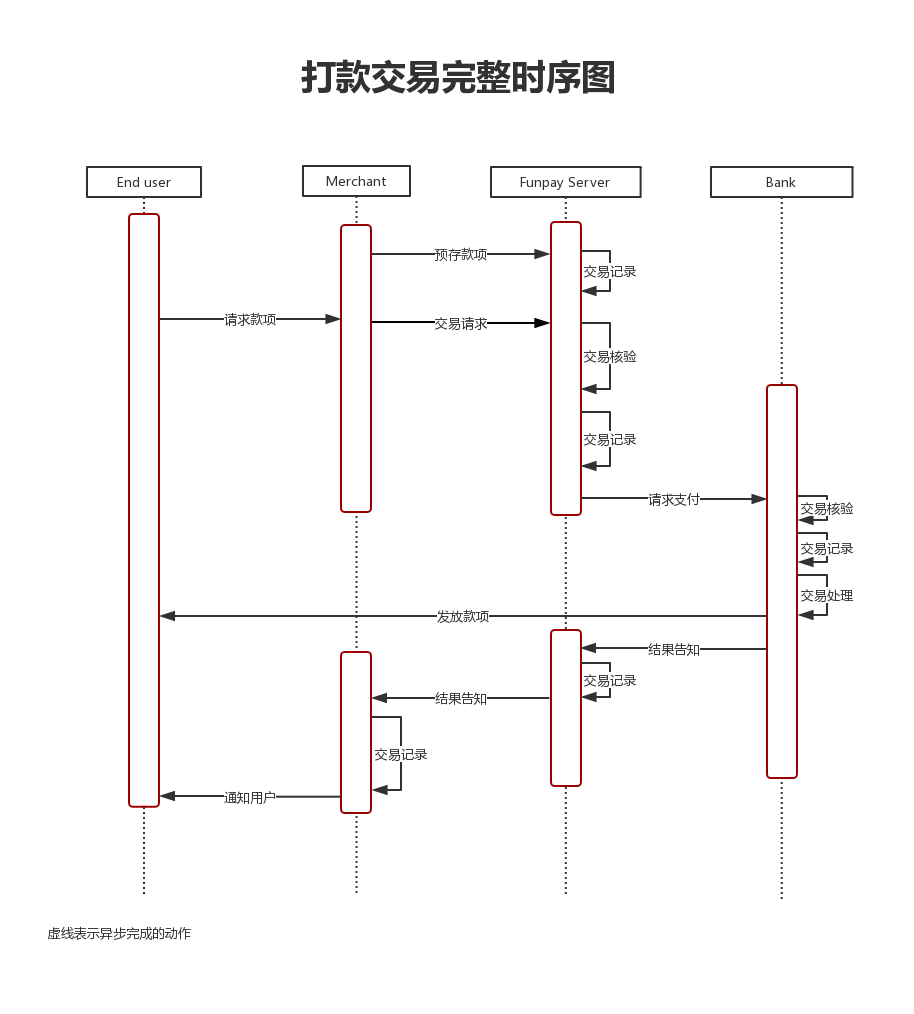
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#更新线下支付码result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID，商家传入，透传回来 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| status | int | Y | 订单状态，详见[订单状态码表](#_状态码表)。 |
| code | string | Y | 需要展示给用户的交易码，用户可以根据此码去便利店进行支付，交易成功后此码不再有效，因此支付结果回调的时候不一定会再次返回此码。 |
| shopLink | url | Y | 便利店地址，商家需要将此链接展示给用户，用户点击后可以用webview打开相关页面，以展示其附近的便利店地图。交易成功后此码不再有效，因此支付结果回调的时候不一定会再次返回此码。 |
| expireDate | string | Y | 过期时间，商家设置，透传回来 |
| userName | string | Y | 用户姓名，商家设置，透传回来 |
| phoneNumber | string | Y | 用户手机号，商家设置，透传回来 |
| returnUrl | string | Y | 回调地址，商家设置，透传回来 |

# *打款功能*

1. 打款交易流程



1. 打款交易接口地址

<https://payment.funpay.asia/fun/transfer/transferMoney>

1. 打款交易接口方法

GET

1. 打款交易所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| clientID | string | N | 用户ID，商户传入，以便区分用户对账平账，商户可以根据自己需要传入终端用户的ID或者合同号等唯一标记，Funpay对此字段不作区分处理，返回结果中亦不进行返回，仅用于结果报表中的检索。商户可以根据自己需求决定是否传入。如果不传入请不要使用此字段。如果带了此字段请务必确保其有值。 | |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| amount | number | Y | 打款金额，单位1VND |
| currency | string | Y | 货币符号，必须为“VND” |
| orderNo | string | Y | 商家自定义订单号，唯一，最长40字符，仅数字字母，不支持特殊字符 |
| returnUrl | url | Y | 回调函数地址，银行打款完成后Funpay会调用此接口以通知商家，不需要URL编码，因为不会直接传输链接地址。Funpay回调此接口的具体说明参见“[打款结果回调](#打款结果回调)”。此参数必须为“http://”开头或者“https://”开头，否则不会触发回调。强烈建议就算您不依赖回调，也填入一个合法的回调地址，可以不做任何逻辑处理，但是方便进一步扩展。 |
| accountName | string | Y | 用户姓名，即收款的银行卡持有者的姓名。需与银行留存信息完全一致。一般为大写英文字母。Funpay作了兼容处理，如用户输入越南文字母，会自动进行转换。但建议用户自行处理完成后再传入Funpay。商家在使用此接口前，可以先通过[核验银行信息接口](#_核验银行卡信息功能)或者其他方法核验相关信息的正确性，以确保打款成功率。此参数为必传参数，Funpay会检测此信息是否与银行留存一致，如不一致或者为空，会拒绝打款。 |
| accountNo | string | Y | 用户银行卡账号或者卡号。 |
| accountType | number | Y | 代表accountNo的类型，0-银行账号（默认值），1-银行卡号 |
| bankLocation | string | Y | 用户银行卡归属国家，目前仅支持“vn” |
| bankNo | number | Y | 用户银行卡归属银行代号，参见“[获取银行列表](#_获取银行列表_2)”。 |
| bankBranchNo | number | Y | 用户银行卡归属银行的支行代号，参见“[获取银行列表](#_获取银行列表_2)”。1.3版本的时候，可以不用再传这个参数，置空即可（即不输入任何值）。 |
| remark | string | Y | 打款的附录语言，会展示给用户，仅支持英文大小字母和数字。最大长度100个字符，含空格，必填，可以为若干个空格键。 |
| transferTime | string | Y | 指定给用户打款的具体日期，立即打款置空即可（置空指不传任何文字，而不是输入NULL） |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |
| isAsync | number | N | 由于本接口是跟Napas的实时同步接口，所以会有比较久的等待时间（大约40s），为了提升效率，所以商户可以用此参数来指明是否同步等待结果实时返回。如果同步返回则本API会等待结果从银行返回后同步返回结果给调用方。而且不管是否已经read timeout，都会触发一次回调通知。如果非同步返回，则本API会先返回创建好的tradeNo，用户可以选择等待回调或者用API来Funpay系统查询。1—非同步返回。用户不设置该值或者设置为非1的值均代表同步返回。默认值为0。 |

1. 打款交易返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#打款Result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| accountName | string | Y | 用户姓名，商家传入，透传回来。 |
| accountNo | string | Y | 用户银行账号/ 卡号，商家传入，透传回来。 |
| accountType | number | Y | 用户accountNo类型，商家传入，透传回来。 |
| bankLocation | string | Y | 用户银行卡归属国家，商家传入，透传回来。 |
| bankNo | number | Y | 用户银行卡归属银行代号，商家传入，透传回来。 |
| bankBranchNo | number | Y | 用户银行卡归属银行的支行代号，商家传入，透传回来。 |
| sendTime | string | Y | 指定给用户打款的具体日期，商家传入，透传回来。 |
| status | number | Y | 订单状态，详见[订单状态码表](#_状态码表)。 |

1. 打款结果回调
2. 打款后回调都会被触发，不论成功失败。
3. 回调的HTTP方法：

POST

1. 回调的参数：

参数表同[打款交易返回值](#打款返回值) ，以JSON数据结构组织。

# 获取银行以及支行列表

1. 接口说明

我们在不停的对接新的银行，所以一直会更多银行在我们的服务支持范围内，为了能及时同步Funpay的最新支持银行，建议合作商家可以定期（建议每周跑一次）来Funpay请求更新最新的银行列表和支行列表。为了方便商家管理，Funpay的对每个银行和支行都设置了一个版本号，其中支行的版本号是方便扩展用，商家可以不用考虑。银行版本号的意思是如果银行改变名称或者新开了更多支行等各种变动，银行版本号也会跟着变动，如果商家发现Funpay的银行版本号比本地的银行版本号更新，那么商家可以更新数据库中的本条目并且重新拉取该银行的支行列表。从本版本开始Funpay不再需要商家提供支行ID，所以ID为“39”以后的银行ID将不可再取得支行列表，合作商家也不必再更新支行列表。

银行也可能存在关闭或者业务变更的风险，所以为了解决这个问题，Funpay在每个银行后面加入了一个银行状态的标志位“status”，来标识我们当前是否能支持对该银行的打款操作。如果该标志位置位，则对该行银行卡的打款会失败。

1. 获取相关银行LOGO

如果需要各个银行的Logo，可以用以下地址+银行ID+“.png“获取。

[https://pic.funpay.asia/bank/](https://pic.funmobi.asia/bank/)

如ID为1的银行对应的银行对应的地址为

[https://pic.funpay.asia/bank/1.png](https://pic.funmobi.asia/bank/1.png)

但是用支行的ID去获取将无法获取到任何图片。这些银行LOGO皆为透明背景图片。

1. 获取银行列表接口地址

<https://payment.funpay.asia/fun/transfer/getBankList>

1. 获取银行列表接口方法

GET

1. 获取银行列表交易所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| bankID | number | Y | 待查询的银行ID，如果查所有银行列表，可以传入0，如果查某银行的支行列表可以传入该银行的ID |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |

1. 获取银行列表返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#打款Result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| count | number | Y | 列表中返回的项目数量 |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| bankID | number | Y | 银行ID，商家传入，透传回来 |
| bankList | array[object] | Y | 订单状态，详见[银行列表数据结构体](#银行列表数据结构)。 |

1. 银行列表数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| bankName | string | Y | 银行或者支行名称，可以展示给用户选择 |
| bankID | number | Y | 银行或者支行ID，与Funpay交互主要使用这个ID |
| version | number | Y | 数据版本号，从1开始，数据越小，版本越老 |
| status | number | Y | 银行支持状态 0-正常支持 1-暂不支持。 |
| abbreviation | string | Y | 银行名称简写 |
| accountSupport | number | Y | 该银行是否支持用银行账号来进行打款。0—不支持，1—支持。友情提醒：之前已经支持的所有39家银行都支持账户打款。 |
| cardSupport | number | Y | 该银行是否支持用银行卡号来进行打款。0—不支持，1—支持。 |

# 核验银行卡信息功能

1. 接口说明

该接口用于为商家核验输入的卡号/账户与银行是否为其本人所有，避免用户输错造成商家损失。

该接口功能为Napas所提供，其结果不受Funpay控制，Funpay亦不对其返回结果负责。

1. 核验银行卡信息接口地址

<https://payment.funpay.asia/fun/transfer/verifyBankInfo>

1. 核验银行卡信息接口方法

GET

1. 核验银行卡信息接口所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| bankNo | number | Y | 待查询的银行ID |
| bankLocation | string | Y | 用户银行归属国家，目前仅支持“vn” |
| accountNo | string | Y | 待查询的银行账户 |
| accountType | number |  | 待查询的银行账户类型，0—accountNo是账号，1—accountNo是卡号 |
| accountName | string | N | 用户姓名，可以传入带音标的越南字符，Funpay会自动对传入的值和Napas下发的值进行比较，并返回比较结果（主要是越南文带音调的字符和不带音调的字符转换困难，这一部分Funpay提供了完整的解决方案）。不是必须，用户可以根据自己需求进行选择是否由Funpay代为比较。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |

1. 核验银行卡信息返回值。以JSON数据结构组织。

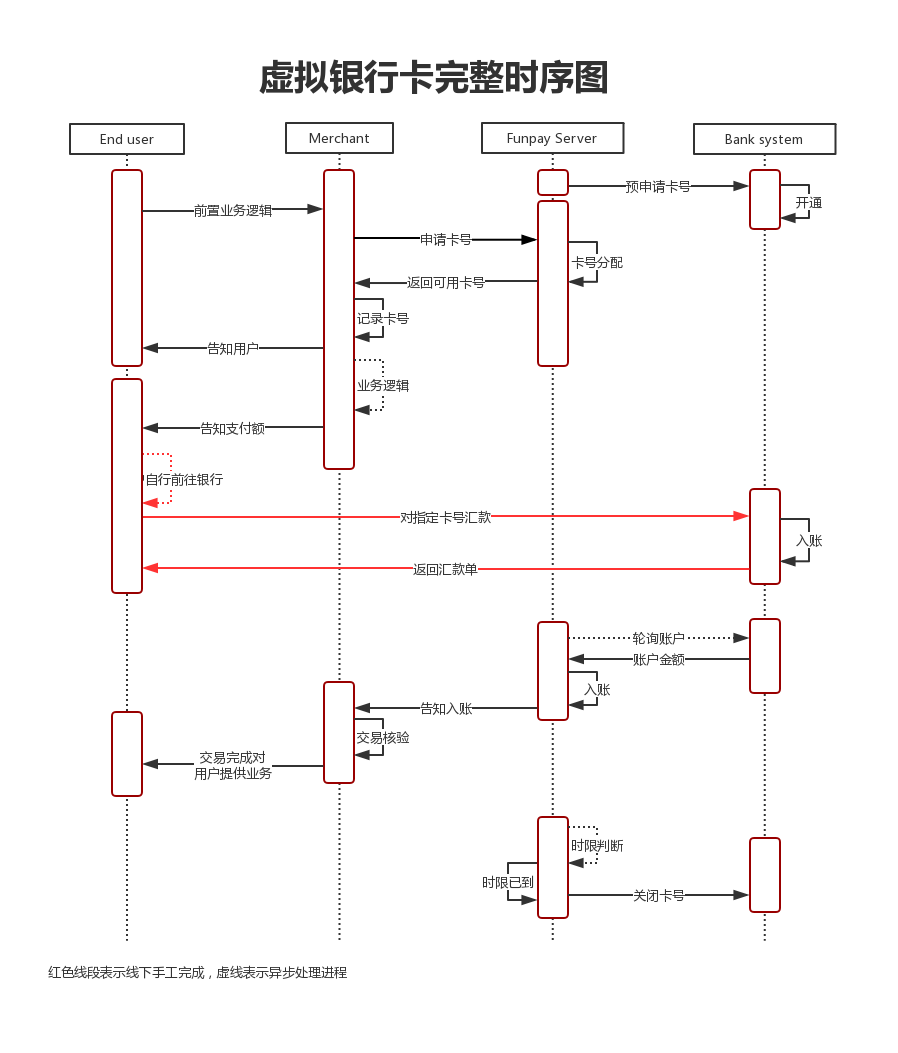
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#核验银行信息result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| bankNo | number |  | 待查询的银行ID，商家传入，透传回来 |
| bankLocation | string | Y | 用户银行归属国家，目前仅支持“vn”，商家传入，透传回来 |
| accountNo | string | Y | 待查询的银行账户，商家传入，透传回来 |
| accountType | number | Y | 待查询的银行账户类型，商家传入，透传回来 |
| accountName | string | Y | 查询得到的用户姓名，即留存在银行的姓名，一般为英文大写字母（注意此处不是透传的商户传入的accountName） |
| verify | number | Y | Funpay的比对结果，0—是同样的姓名，-1—用户未传入姓名进行比对，1—不是同样的姓名 |
| status | number | Y | 0—查询成功，非0—查询失败。此处查询失败指Napas返回账户非法或者未检索到结果，商家可以多次尝试。 |

# 虚拟银行卡功能

1. 交易流程图



1. 交易流程说明
2. FunPay已经在合作银行注册了大量的虚拟银行卡（Virtual Card 以下简称VC），需要商家根据自己的需要跟FunPay申请VC的使用权，并指定该卡的使用期限以及期间的交易通知地址。
3. 申请到VC后需要把该银行卡号告知用户，同时需要告知用户以下信息：

VIB虚拟卡

i：收款单位： CTY TNHH CONG NGHE FINPOINT VN

ii：收款银行名称： VIB

iii：支行名称： HOI SO CHINH

VietinBank（以下简称VTB）虚拟卡

i：收款单位： CONG TY TNHH CONG NGHE FUNMOBI VIET NAM

ii：收款银行名称： VietinBank

iii：支行名称： Chuong Duong

1. 由于成本和收费问题的原因，用户每次存入钱都需要支付一定的服务费用，需要合作伙伴告知用户这笔费用的存在。
2. 用户凭借2）中提到的信息去银行或者通过网银进行转账即可。
3. Funpay会轮询该VC的交易情况，发现进账后会通过回调通知到商家入账情况。
4. 使用时限到了之后，或者用户完成交易关闭卡号以后，Funpay会关闭该VC，并不再回调相关交易结果。
5. 关于两家银行虚拟卡的说明
6. 由于VTB银行原因，VTB虚拟卡仍存在一些问题，暂不具备对商家开放使用的条件。Funpay正在跟VTB银行继续联调修正中，待开放条件具备后，Funpay会通知到商家使用。
7. 由于越南虚拟卡的局限性，暂时无法支持在ATM对虚拟卡进行存款操作，暂时也无法支持Napas的7\*24快速转账通道，我们在努力推动银行进行这个问题的优化 。
8. 建议使用用户身份证号作为VIB VC的银行卡号，一人一号，终身有效，这样可以有效避免卡号重复，同时大概率降低用户的输错机会。
9. VIB VC支持创建/激活/失活/释放虚拟卡，未被创建/失活/已释放的虚拟卡不可以再进行存入交易。在用户创建虚拟卡的时候Funpay会自动激活，不需要商家重复调用。
10. 商家可以根据自己需要选择某一种VC提供给用户，也可以让用户自行选择方便使用的VC归属银行，或者同时申请两个银行的VC，让用户自行选择实际使用的VC。
11. VIB虚拟卡释放后，必须重新申请才能再次使用，否则无法存入金额。VIB虚拟卡只能在创建的时候绑定用户姓名，因此，如果更改用户姓名只能释放虚拟卡并重新申请。
12. VIB虚拟卡被关闭（失活）后，必须重新激活才能再次使用，否则无法存入金额。所以建议如果临时不需要存入交易的时候，可以关闭（失活）虚拟卡，如果用户解除跟我们的合作，不再使用虚拟卡，可以释放虚拟卡，以待重新申请。
13. 不管是哪种虚拟卡，都只能获取到交易的入账时间，不能获得该交易的发起时间，因为交易发起银行不会告知收款银行交易实际发起时间 。VIB虚拟卡入账时间只能具体到日，暂时无法精确到时分秒。
14. 由于银行系统原因，非工作时间（包含节假日和每日的非工作时间）对虚拟卡的存款只能在工作时间才能处理入账，所以商家只能以实际入账时间来判断用户存款并进行清算，所以需要提醒用户规划好其时间，避免因非工作时间存款导致的损失。
15. 申请使用 VC
16. 接口地址

[https://payment.funpay.asia/fun/payment/virtualCard/create](https://payment.funmobi.asia/fun/payment/virtualCard/create)

1. 使用的方法

POST

1. 使用的Content-Type

application/json (可以带，也可以不带“;charset=utf-8”)

1. 所需参数（以JSON结构体传输）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| feeID | number | Y | 计费点ID，funpay后台分配 |
| clientID | string | N | 用户ID，商户传入，以便区分用户对账平账，商户可以根据自己需要传入终端用户的ID或者合同号等唯一标记，Funpay对此字段不作区分处理，返回结果中亦不进行返回，仅用于结果报表中的检索。商户可以根据自己需求决定是否传入。如果不传入请不要使用此字段。如果带了此字段请务必确保其有值。 | |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| amount | number | Y | 用户的订单额，单位1VND |
| currency | string | Y | 用户的订单货币单位，唯一值VND |
| orderNo | string | Y | 商户自行创建的订单号，唯一，最长40字符，仅数字字母，不支持特殊字符 |
| expireDate | string | Y | 商户自行指定的过期时间，格式yyyyMMdd，以河内时间为准。过了该日后的第二天凌晨就会回收该虚拟账户，用户无法存入金额。 如果不需要该虚拟卡号过期，则置空，不传任何值。 对于VTB虚拟卡必须设置此有效期，且不得超过一个月，否则创建时会返回错误。对于VIB虚拟卡，建议将此参数设置为空，以示永久有效。 |
| returnUrl | string | Y | 回调地址，Funpay将以此地址通知用户进来的每一次汇款。此参数必须为“http://”开头或者“https://”开头，否则不会触发回调。强烈建议就算您不依赖回调，也填入一个合法的回调地址，可以不做任何逻辑处理，但是方便进一步扩展。 |
| bankType | string | Y | 银行类型，默认值为“VIB“，代表VIB银行 。可用值“VIB”—VIB银行，“VTB” —VTB银行。 |
| accountNo | string | Y | 用户身份证号，会被用于作为银行卡号，不超过15位数字，不足15位的部分，Funpay会自动用0在前面补齐。如相同的卡号已经创建，则Funpay会返回创建失败。在bankType为“VTB”时该参数失效，传入任何值均无实际意义，只作为计算sign使用。 |
| userName | string | Y | 用户姓名，会传入给银行，作为交易的区分标记。如输入含有音调的字母，Funpay会自动转换为不带音调的字母传入给银行。不支持数字和特殊字符。用户对该账户存入的时候输入的不是这个姓名，仍然是Funpay公司名。在bankType为“VTB”时该参数失效，传入任何值均无实际意义，只作为计算sign使用。 |

1. 返回值（以JSON数据结构组织）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[申请VC返回值的result数据结构](#创建VC的result数据结构)”与 |

1. 返回值中的result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID，funpay后台分配 |
| orderNo | string | Y | 商户自行创建的订单号，唯一，最长40字符，仅数字字母，不支持特殊字符 |
| expireDate | string | Y | 商户自行指定的过期时间，格式yyyyMMdd，以河内时间为准。 |
| amount | number | Y | 用户的订单额，单位1VND，商家传入，透传回来 |
| currency | string | Y | 用户的订单货币单位，唯一值VND |
| accountNo | string | Y | Funpay为该用户分配的 VC号码。 Funpay为该用户分配的VIB VC号码，格式为：FUN（三位固定字母）+四位商家流水号（数字，即为merchantID）+十五位用户流水号（数字，即为传入的参数accountNo，不足十五位的，Funpay自动在前面补齐） |
| accountName | string | Y | 银行户主名称，固定值为VIB：“CTY TNHH CONG NGHE FINPOINT VN”，VTB：“CONG TY TNHH CONG NGHE FUNMOBI VIET NAM”，商家需要将此信息展示给用户 |
| bankName | string | Y | 银行名称，固定值为“VIB”或者“Vietinbank”，商家需要将此信息展示给用户 |
| branchBankName | string | Y | 银行支行名称，固定值为“HOI SO CHINH”（VIB银行）或者“Chuong Duong”（VTB银行），商家需要将此信息展示给用户 |
| bankLink | string | Y | 银行地图信息，固定值为<https://www.vib.com.vn/wps/portal/vn/atm>（VIB银行）或者  <https://www.vietinbank.vn/vnetwork/>（VTB银行），支持使用所有银行还款，但在VC所属操作会有更低的费率，如果需要，商家可以把银行地图展示给用户。 |
| serviceFee | number | Y | 用户服务费，用户每次存入钱都会扣取这个金额的服务费，需要预先告知客户。单位：1VND |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |

1. VC交易结果回调
2. 回调地址

商家传入的returnUrl

1. 使用的方法

POST

1. 使用的Content-Type

application/json;charset=utf-8

1. 回调参数（以JSON数据结构组织）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#创建VC的result数据结构)” |

1. 回调参数中的result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID，funpay后台分配 |
| orderNo | string | Y | 商户自行创建的订单号，唯一，最长40字符，仅数字字母，不支持特殊字符 |
| tradeNo | string | Y | 由Funpay创建的交易号，每次还款此tradeNo均会变更。 |
| expireDate | string | Y | 商户自行指定的过期时间，格式yyyyMMdd，以河内时间为准。 |
| currency | string | Y | 用户的订单货币单位，唯一值VND |
| accountNo | string | Y | Funpay为该用户分配的VC号码 |
| accountName | string | Y | 银行户主名称，固定值为VIB：“CTY TNHH CONG NGHE FINPOINT VN”，VTB：“CONG TY TNHH CONG NGHE FUNMOBI VIET NAM”，商家需要将此信息展示给用户 |
| amount | number | Y | 订单金额，商家传入，透传回来 |
| bankName | string | Y | 银行名称 |
| branchBankName | string | Y | 银行支行名称 |
| bankLink | string | Y | 银行地图信息 |
| purchaseAmount | number | Y | 用户的存入金额，已扣除相关serviceFee，单位1VND |
| purchaseCurrency | string | Y | 用户的存入金额单位，固定值，VND |
| purchaseTime | string | Y | 用户的款项的实际入账时间，格式yyyyMMddHHmmss。以河内时间为准。VIB虚拟卡入账时间只能具体到日，暂时无法精确到时分秒。VTB银行会返回精确的时分秒。 |
| remark | string | Y | 银行返回的打款备注 |
| serviceFee | number | Y | 用户服务费，用户每次存入钱都会扣取这个金额的服务费，需要预先告知客户。单位：1VND |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |

1. 释放 VC
2. 特别说明

对于VIB虚拟卡，只有释放后才能绑定新的用户名，且被释放的虚拟卡，无法被再次存入款项，直至再次申请并激活。对于VTB银行卡，仅起到交易完成释放卡号，以给其他交易使用的目的。

对于VIB虚拟卡，释放的使用场景是该用户已经注销账户，不再使用商家该产品，商家需要回收此虚拟卡，以便重新赋予其他用户使用。

1. API地址

<https://payment.funmobi.asia/fun/payment/virtualCard/destory>

1. 使用的方法

POST

1. 使用的Content-Type

application/json (可以带，也可以不带“;charset=utf-8”)

1. 所需参数（以JSON结构体传输）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** | |
| merchantID | number | Y | 商家ID，funpay后台分配 | |
| businessID | number | Y | 业务ID，funpay后台分配 | |
| feeID | number | Y | 计费点ID，funpay后台分配 | |
| timestamp | number | Y | UNIX系统时间戳，当前时间 | |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) | |
| orderNo | string | N | 商户自行创建的订单号，唯一，最长40字符，仅数字字母，不支持特殊字符。与accountNo二者至少存在一个。 | |
| accountNo | string | N | 需要释放的银行卡号，与orderNo二者至少存在一个。 | |
| bankType | string | N | 银行类型，默认值为“VIB“，代表VIB银行，可传可不传。可用值“VIB”—VIB银行，“VTB” —VTB银行。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 | |

1. 接口返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | NULL |

1. 激活/更新 VC
2. API说明

该接口仅对VIB虚拟卡有效，对于VTB虚拟卡，调用该接口无任何意义。

在虚拟申请后Funpay会自动激活，不需要商家重新申请激活。该接口仅对状态为禁用/失活的虚拟卡有效，对未创建、已释放的虚拟卡无效。

对于VIB虚拟卡激活的使用场景是，商家禁用了虚拟卡后重新启用虚拟卡，以方便用户再次使用。

对于已经激活的虚拟卡，调用此接口，可以更新虚拟卡信息。

1. API地址

<https://payment.funmobi.asia/fun/payment/virtualCard/enable>

1. 使用的方法

POST

1. 使用的Content-Type

application/json (可以带，也可以不带“;charset=utf-8”)

1. 所需参数（以JSON结构体传输）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** | |
| merchantID | number | Y | 商家ID，商家传入，透传回来 | |
| businessID | number | Y | 业务ID，商家传入，透传回来 | |
| feeID | number | Y | 计费点ID，funpay后台分配 | |
| timestamp | string | Y | UNIX系统时间戳 | |
| version | string | Y | 版本号，固定值“1.3” | |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) | |
| orderNo | string | N | 商户自行创建的订单号，唯一，最长40字符，仅数字字母，不支持特殊字符。 如未激活则必须传入。如已激活且不需更新，则不需要传入。如已激活并传入新值，代表更新此参数 | |
| expireDate | string | N | 商户自行指定的过期时间，格式yyyyMMdd，以河内时间为准。如已激活且不需更新，则不需要传入。如未激活则必须传入。可以置空，代表永久有效。 | |
| amount | number | N | 用户的订单额，单位1VND，商家传入，透传回来，如已激活且不需更新，则不需要传入。如未激活则必须传入。 | |
| currency | string | N | 用户的订单货币单位，唯一值VND。如已激活且不需更新，则不需要传入。如未激活则必须传入。 | |
| accountNo | string | Y | 需要更新或者激活的银行卡号 | |
| returnUrl | string | N | 通知结果的回调地址。如已激活且不需更新，则不需要传入。如未激活则必须传入。 | |
| bankType | string | N | 银行类型，默认值为“VIB“，代表VIB银行，可传可不传。可用值“VIB”—VIB银行，“VTB” —VTB银行。 |
| clientID | string | N | clientID。如已激活且不需更新，则不需要传入。如未激活则必须传入。 | |

1. 返回值

[与申请使用VC接口返回值相同](#申请使用VC接口返回值)

1. 禁用/失活虚拟卡
2. API说明

该接口仅对VIB虚拟卡有效，对于VTB虚拟卡，调用该接口无任何意义。

该接口仅对已激活的虚拟卡有效。对于禁用/未创建/已释放的虚拟卡无任何意义。

对于VIB虚拟卡禁用的使用场景是，用户一段时间内不需要再次对商家进行支付，为了避免用户误输入，商家可以禁用虚拟卡。

1. API地址

<https://payment.funmobi.asia/fun/payment/virtualCard/disable>

1. 使用的方法

POST

1. 使用的Content-Type

application/json (可以带，也可以不带“;charset=utf-8”)

1. 所需参数（以JSON结构体传输）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** | |
| merchantID | number | Y | 商家ID，商家传入，透传回来 | |
| businessID | number | Y | 业务ID，商家传入，透传回来 | |
| feeID | number | Y | 计费点ID，funpay后台分配 | |
| timestamp | string | Y | UNIX系统时间戳 | |
| accountNo | string | Y | 需要关闭的银行卡号 | |
| version | string | Y | 版本号，固定值“1.3” | |
| bankType | string | N | 银行类型，默认值为“VIB“，代表VIB银行，可传可不传。可用值“VIB”—VIB银行，“VTB” —VTB银行。 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) | |

1. 返回值

[与释放VC接口返回值相同](#释放VC接口返回值)

# 交易结果查询功能

1. 注意事项

用于交易完成后，给接入商户查询订单状态，以便核验，可选功能。

1. 支付结果查询
2. 接口地址

<https://payment.funpay.asia/fun/payment/check>

1. 使用的方法

GET

1. 所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| tradeNo | string | Y | 由Funpay创建的交易号，在创建订单或者返回结果的时候都会返回这个交易号，Funpay创建，唯一。可以传空，如果传入数据，将会作为有效值与orderNo一起进行检索。如果orderNo与tradeNo同时为空将会直接返回错误。 |
| orderNo | string | Y | 由商家创建的交易号，唯一。可以传空，如果传入数据，将会作为有效值与tradeNo一起进行检索。如果orderNo与tradeNo同时为空将会直接返回错误。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |

1. 返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#支付结果查询result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算 |
| name | string | Y | 计费点名称 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| purchaseTime | string | Y | 实际支付时间，如尚未支付，会设置为空，格式为yyyyMMddHHmmss |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| status | number | Y | 订单状态，详见[订单状态码表](#_状态码表)。 |

1. 打款结果查询
2. 接口地址

<https://payment.funpay.asia/fun/transfer/check>

1. 接口方法

GET

1. 所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| tradeNo | string | Y | 由Funpay创建的交易号，在创建订单或者返回结果的时候都会返回这个交易号，Funpay创建，唯一。可以传空，如果传入数据，将会作为有效值与orderNo一起进行检索。如果orderNo与tradeNo同时为空将会直接返回错误。 |
| orderNo | string | Y | 由商家创建的交易号，唯一。可以传空，如果传入数据，将会作为有效值与tradeNo一起进行检索。如果orderNo与tradeNo同时为空将会直接返回错误。 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |

1. 返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#打款结果查询result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| status | number | Y | 订单状态，详见[订单状态码表](#_状态码表)。 |
| accountName | string | Y | 用户账户对应姓名 |
| accountNo | string | Y | 用户银行账户 |
| accountType | number | Y | 用户银行账户类型，商家传入，透传回来 |
| bankLocation | string | Y | 用户银行归属国家，目前仅支持“vn” |
| bankNo | string | Y | 打款银行ID |
| bankBranchNo | string | Y | 打款银行支行ID |
| sendTime | string | Y | 打款时间，格式为yyyyMMddHHmmss |

1. 虚拟卡交易查询
2. 接口地址

<https://payment.funpay.asia/fun/payment/virtualCard/check>

1. 接口方法

GET

1. 所需参数（所有参数需要以“[API加密规则](#_API加密规则_3)”提供的方案整合为唯一参数“param”）

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，funpay后台分配 |
| businessID | number | Y | 业务ID，funpay后台分配 |
| timestamp | number | Y | UNIX系统时间戳，当前时间 |
| version | number | Y | 此处是固定值“1.3”（注意，实际使用时不需要加引号），代表API版本V1.3 |
| sign | string | Y | 签名，[计算规则见URL加密规则](#_API加密规则) |
| tradeNo | string | N | 需要查询的tradeNo。可以与orderNo、accountNo一起指定，组成并的关系。也可以单独指定，但是tradeNo、orderNo、accountNo三者至少其中一个存在。 |
| orderNo | string | N | 需要查询的orderNo。可以与tradeNo、accountNo一起指定，组成并的关系。也可以单独指定，但是tradeNo、orderNo、accountNo三者至少其中一个存在。 |
| accountNo | string | N | 需要查询的accountNo。可以与tradeNo、orderNo一起指定，组成并的关系。也可以单独指定，但是tradeNo、orderNo、accountNo三者至少其中一个存在。 |
| bankType | string | Y | 银行类型，默认值“VIB”，必须指定。 |
| pageSize | number | Y | 查询的每页容量，默认值10，最大值100，超过100视为100。 |
| pageNum | string | N | 查询的页码，默认值0，第一页为0。 |

1. 返回值。以JSON数据结构组织。

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| code | number | Y | 结果码，具体数值请参考“[结果返回码表](#_结果返回码表)” |
| msg | string | Y | 对结果码的说明，辅助调试接口用 |
| result | object | Y | 实际结果内容，只有在code为10000的时候此参数才有实际意义。该结构体实际内容参考下表“[result数据结构](#打款结果查询result数据结构)” |

1. result数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| cnt | number | Y | 总条目数 |
| pageCnt | string | Y | 总页码数 |
| pageNum | string | Y | 当前页码数，商户传入，透传回来 |
| pageSize | string | Y | 每页容量，商户传入，透传回来 |
| nextPageFlag | boolean | Y | 是否还有下一页，true—有，false—没有 |
| sign | object | Y | 签名，[计算规则见URL加密规则](#_API加密规则)，此处回传以进行数据校验，仅result中数据参与MD5计算。且“transactions”不参与签名运算。 |
| transactions | array[object] | Y | JSON数组，罗列所有符合条件的交易。此参数不参与sign的运算。该结构体实际内容参考下表“[transaction数据结构](#查询虚拟卡的transaction数据结构)”。 |

1. transaction数据结构

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **是否必须** | **意义** |
| merchantID | number | Y | 商家ID，商家传入，透传回来 |
| businessID | number | Y | 业务ID，商家传入，透传回来 |
| feeID | number | Y | 计费点ID，商户传入，透传回来 |
| orderNo | string | Y | 商家自定义订单号，商家传入，透传回来 |
| tradeNo | string | Y | 支付网关自己创建的交易流水号，对账使用，最长40位，仅包含数字字母 |
| amount | number | Y | 支付金额，商家传入，透传回来 |
| currency | string | Y | 货币符号，商家传入，透传回来 |
| purchaseAmount | number | Y | 用户实际支付金额 |
| purchaseCurrency | string | Y | 用户支付货币符号，固定值“VND” |
| purchaseTime | string | Y | 用户入账时间，以河内时间为准，格式为yyyyMMddHHmmss |
| remark | string | Y | 实际支付打款备注 |
| accountNo | string | Y | 实际还款银行虚拟账户 |
| bankName | string | Y | 用户实际支付银行名称“VIB”或者“VietinBank” |
| bankBranchName | string | Y | 用户实际支付银行支行名称 |
| accountName | string | Y | 账户归属户主姓名 |
| bankLink | string | Y | 银行地图链接 |
| serviceFee | number | Y | 用户服务费 |

# 自动对账功能

# 结果返回码表

|  |  |  |
| --- | --- | --- |
| **code** | **msg** | **描述** |
| 10000 | the request is succeed. | 请求成功，一切正常 |
| 40000 | the param is wrong. | 参数错误 |
| 40001 | the encrypt is wrong. | 加密错误，取到的值与预期不符 |
| 40004 | the request is not found. | 请求资源（如交易号、订单号）不存在或者输入参数有问题 |
| 40009 | the product status is illegal. | 产品状态异常，如果是打款请检查备注是否已超过100个字符，如果是虚拟卡请检查该卡状态是否符合预期 |
| 50000 | the connection error. | 系统错误 |
| 50001 | the connection error. | 系统错误 |
| 50002 | the request is rejected. | 请求被拒绝，一般由于系统错误导致 |
| 50006 | the request is rejected. | 请求被拒绝，无API使用权限 |
| 50009 | the cardNo is not exist. | 所查询的银行不存在 |
| 60001 | the balance is insufficient. | 余额不足 |
| 60005 | the transaction is duplicated. | 交易重复 |
| 60050 | The user does not exist. | 用户不存在或者虚拟卡不存在 |

# 订单状态码表

|  |  |
| --- | --- |
| **值** | **描述** |
| 0 | 交易成功，即用户已完成支付或者银行处理完打款。 |
| 1 | 银行拒绝交易（账户或者银行卡被锁定） |
| 3 | 银行卡过期 |
| 4 | 不符合限定条件（OTP错误、有限期不对、CVC、CVV错误） |
| 5 | 银行无响应 |
| 6 | 与银行通讯失败 |
| 7 | 资金不足 |
| 8 | 检验码错误 |
| 9 | 银行账户/卡号、姓名或者交易类型核对不通过 |
| 10 | 其他未定义错误 |
| 11 | 卡校验成功 |
| 12 | 交易额超出限制 |
| 13 | 用户未开通网银，请用户与发卡银行联系 |
| 14 | OTP错误 |
| 15 | 密码错误 |
| 16 | 银行账户与持有者不对应，打款和线上支付有效 |
| 17 | 卡号错误 |
| 18 | 卡的签发日期输入错误 |
| 19 | 卡的有效日期输入错误 |
| 20 | 交易失败 |
| 21 | OTP超时 |
| 22 | 交易校验失败，请检查集成文档 |
| 23 | 卡不被允许支付 |
| 24 | 卡交易额超过限定 |
| 25 | 卡交易额超过限定 |
| 26 | 交易处理中，如等待用户支付或者等待银行处理打款 |
| 27 | 身份验证信息错误 |
| 28 | 交易日期超过限定 |
| 29 | 交易失败，需要与银行确认原因 |
| 30 | 交易失败，交易额小于限定 |
| 31 | 未找到订单 |
| 32 | 非支付用的订单 |
| 33 | 订单重复 |
| 34 | 打款交易失败 |
| 35 | 订单已过期 |

# 