OpenPay API Reference

1 修订记录

修改内容	修改原因	修改时间
创建文档	新建	2023-10-11
修复 payment_intent 和 disburse 的 create 接口中 channel 和 fund_channel 的枚举值	调整渠道的枚举定义	2023-10-17
补充支付和代付查询列表接口的说明，补充回调通知的说明	查询列表的接口之前没有明确有介绍	2023-10-18
去掉代付目标银行账号需要 RSA 加密的要求	用处不大，反而增加对接复杂度	2023-10-18

2 一些约定

2.1 通用参数

字段	命名规则	取值规则
币种	以 currency 结尾	币种编码遵照 ISO4217，用 Alphabetic Code，都用小写字母，长度为 3
金额	以 amount 结尾	基于基本货币单位的金额，精确到小数点后两位

时间	以 at 结尾	UTC 时间，精确到毫秒
比率	以 rate 结尾	表达百分比，去掉百分号的数字部分，具体精确到小数点后的位数，在具体字段里说明

2.2 应答消息

API 采用 RESTful 风格，应答消息中，用 http status code 表示请求的处理情况，如果是成功，则为 200，消息体为具体业务对象；如果是失败，不同的 status code 表达不同的失败原因分类，在消息体中，用 code 表达具体错误码，message 表达详细错误原因

```
{
    "code": "payment_method_unsupported"   //字符串形式，易读的错误码
    "message": "CoinsPay was closed"        //更具体的错误信息，可能会带上链接到商户 dashboard 的详细说明的 url
}
```

3 错误码定义

OnePay 采用 HTTP 应答的 status code 表示 API 请求的成功或失败，这样方便运维层和业务层的分级监控。具体的错误信息在消息体中表示，并且，OnePay 放弃了常用的数字表达的错误码，而采用更加易读的字符串形式的错误码。

3.1 HTTP 状态码汇总

HTTP Code	**表达的含义**
200	请求处理成功

201	创建成功
400	无效请求，缺少关键的参数
401	鉴权校验失败
403	权限不满足，可能因为业务未开通
404	请求的资源不存在，比如发起退款时对应的支付记录不存在
409	用户尝试创建的资源已存在（比如幂等的接口被重复调用）
429	重复的请求过于频繁
500	服务器未知错误

3.2 消息体的错误码

Code	**表达的含义**
authorization_failed	鉴权失败
sign_validate_failed	验签失败
no_privilege	对访问的资源没有权限
unsupported_payment_method	支付方式类型未开通或者不支持的支付方式
balance_insufficient	代付时余额不足
transaction_expired	交易超时
invalid_bank_account_no	无效的银行账户号

4 鉴权

对商户发起交易请求，采用 bearer token 的方式鉴权。Onepay 会分配给客户 secret key，不同环境 secret key 串会是：sk_test_xxxxxxx 和 sk_prd_xxxxx

注：为了给客户对接和测试接口时提供更好的易用性。我们对交易请求并没有采用 HMAC 的鉴权方式，参考 Stripe API 采用了 bearer token 的认证方式。通过 https 完全可以防消息篡改，此外也有 IP 白名单机制做防护。

5 API 列表

5.1 支付

5.1.1 支付订单(PaymentIntent)

标记 required 的字段表示在请求时必须有

字段名	说明	类型
id	支付订单号，py_xxxxxxx	string
account_id	商户账户 ID	string
ref_trade_id required	关联的商户侧交易 ID	string
amount required	支付金额，基于常用单位	string
currency required	支付币种	string

fee_amount			手续费金额，基于常用单位	string
exchange_rate			汇率，买入结算币种单位金额需要的支付币种的金额	string
status			订单状态，包含以下值： processing 表示支付发起成功，未完成支付扣款 succeeded 表示支付成功 failed 支付发起失败或者支付失败	string
payment_method required				object
	type required		支付方式类型： card direct_debit	string
	direct_debit		如果 type 为 direct_debit，则此项为必须	object
		channel	jcpay_qrcode	string
	card		如果 type 为 card，则此项为必须	object
		brand	卡品牌	string

				mastercard unionpay visa	
		country		卡所属国家	string
	billing_details			账单明细	object
		address		账单地址	object
			city	城市	string
			country	国家	string
			line1	街道、公司名等	string
			line2	单元、楼号等	string
			postal_code	邮政编码	string
			state	州或省	string
		email		卡持有者邮箱	string
		name		卡持有者姓名	string
		phone		卡持有者电话	string
customer				支付用户信息	object
	first_name				string
	mid_name				string
	last_name				string

	email			string
	phone			string
	ip			string
	address		地址信息	object
		city	城市	string
		country	国家	string
		line1	街道、公司名等	string
		line2	单元、楼号等	string
		postal_code	邮政编码	string
		state	州或省	string
available_on			资金到账时间	UTC time
Description required			因为什么而支付的描述信息	string
metadata			请求附带的信息	string
return_url			返回 URL，信用卡支付涉及 3DS 会用到	string
callback_url			回调 URL，如果请求时传了，则使用请求消息中的，否则用配置的 callback url	string

fail_reason	失败原因	string
trans_at	下单时间	timestamp

5.1.2 创建支付订单

```
POST /api/v1/payment_intent
```

5.1.2.1 请求示例

```
{
    "ref_trade_id": "商家订单的 ID",
    "amount": "10.00",
    "currency": "php",
    "customer": {
        "first_name": "wowo",
        "last_name": "totot",
        "email": "wwwd@sina.com"
    },
    "payment_method": {
        "type": "direct_debit",
        "direct_debit": {
            "channel":"jcpay_qrcode"
            }
    },
    "description": "AbC"
}
```

5.1.3 获取一个支付订单

```
GET /api/v1/payment_intent/{payment_intent_id}
```

5.1.4基于商户订单号获取支付订单

GET /api/v1/payment_intent/ref_trade_id/{ref_trade_id}

5.1.5查询支付订单列表

GET /api/v1/payment_intent

参数有：

- start, end：交易时间范围的起止时间，取值为时间戳。可以不传值，也可以只给一个参数传值。
- status：按这个订单状态值查询订单。
- sort_type: desc/asc ,表示按交易时间倒序还是正序排序查询结果。如果不传，默认为倒序。
- page_size：每页的记录数，默认为 10 条，最大 30 条。
- start_after：查询某条记录之后的一页。每次查询返回的 start_after 对应的是当页最后一条记录，把返回消息中的 start_after 值作为参数再次查询，则可以查到下一页的记录，has_more 表示是否还有下一页。
- end_before：查询某条记录之前的一页。每次查询返回的 end_before 对应的是当页第一条记录，把返回消息中的 end_before 值作为参数再次查询，则可以查到前一页的记录，如果返回结果中 end_before 有值有表示还有前一页。

返回信息示例如下：


```
{
    "has_more": true,
    "sort_type": "desc",
    "page_size": 2,
    "start_after":
"eyJzdGFydF9hZnRlciI6InB5XzJXblY4S01DMXZNSkRDM1Iwbk00TVFyOWF6diIsImVuZF9iZWZvcmUiOiIiLCJwY
WdlX3NpemUiOjMsInBhZ2luZ190aW1lIjoxNjk3NTk4ODk2MTU1fQ==",
    "data": [
```

```
        {
            "id": "py_2WnasXLyGvk3NvwXAyw7aXD2iDF",
            "account_id": "acct_juancash",
            "ref_trade_id": "391",
            "amount": "10.00",
            "currency": "php",
            "fee_amount": "0.38",
            "exchange_rate": "0",
            "description": "AbC",
            "customer": {
                "first_name": "wowo",
                "last_name": "totot"
            },
            "trans_at": "1697370998000",
            "status": "succeeded",
            "object": "payment_order"
        },
        {
            "id": "py_2WnV8KMC1vMJDC3R0nM4MQr9azv",
            "account_id": "acct_juancash",
            "ref_trade_id": "390",
            "amount": "10.00",
            "currency": "php",
            "fee_amount": "0.38",
            "exchange_rate": "0",
            "description": "AbC",
            "customer": {
                "first_name": "wowo",
                "last_name": "totot"
            },
            "trans_at": "1697368120000",
            "status": "succeeded",
            "object": "payment_order"
        }
    ]
}
```


5.2 代付

5.2.1 代付订单(DisburseOrder)

标记 required 的字段表示在请求时必须有

字段名	说明	类型
id	支付订单号，dib_xxxxxxx	string
account_id	商户账户 ID	string
ref_trade_id required	关联的商户侧交易 ID	string
amount required	支付金额，基于常用单位	string
currency required	支付币种	string
fee_amount	手续费金额，基于常用单位	string
status	订单状态，包含以下值： processing 处理中，代付发起成功，资金在途 succeeded 代付成功 failed 代付发起失败或代付处理失败	string
fund_channel required	代付资金渠道类型： bdo bpi rcbc	string

	union_bank gcash grab_pay paymaya	
description	下单时的描述信息	string
bank_account_no required	银行账户号	String
bank_code	银行编码	string
bank_account_name	银行账户名	string
metadata	请求附带的信息	string
fail_reason	失败原因	string
callback_url	回调 URL，如果请求时传了，则使用请求消息中的，否则用配置的 callback url	string
fail_reason	失败原因	string
trans_at	下单时间	timestamp

5.2.2 创建代付订单

```
POST /api/v1/disiburse
```

5.2.2.1 请求示例

```
{
    "ref_trade_id": "商家的订单 ID",
    "amount": "15.00",
    "currency": "php",
    "fund_channel": "bpi",
    "bank_account_no": "12345678910",
    "bank_account_name": "Tim"
}
```

5.2.3获取一个代付订单

```
GET /api/v1/disiburse/{disburse_order_id}
```

5.2.4基于商户订单号获取代付订单

```
GET /api/v1/disiburse/ref_trade_id/{ref_trade_id}
```

5.2.5查询代付订单列表

```
GET /api/v1/disiburse/

//查询参数与查询支付订单列表相同
```

返回示例如下：

```
{
    "has_more": true,
    "sort_type": "desc",
    "page_size": 2,
    "start_after":
"eyJzdGFydF9hZnRlciI6ImRpYl8yV3F4Rk5DQ2tWcTRQZmhpWGxwV1NTbmJRZFYiLCJlbmRfYmVmb3JlIjoiIiwic
GFnZV9zaXplIjoyLCJwYWdpbmdfdGltZSI6MTY5NzYwMTM5OTIxOH0=",
```

```
    "data": [
        {
            "id": "dib_2Wryu7iYQMx5KIEBaZr3y0fYNmx",
            "ref_trade_id": "30007",
            "amount": "5.00",
            "currency": "php",
            "status": "accounted",
            "bank_account_name": "taoli",
            "bank_account_no": "09087769164",
            "account_id": "acct_2WoHFSjAh1eaRpVH72RsOc1OyCw",
            "trans_at": "1697503493000",
            "object": "disburse_order"
        },
        {
            "id": "dib_2WqxFNCCkVq4PfhiXlpWSSnbQdV",
            "ref_trade_id": "10030",
            "amount": "5.00",
            "currency": "php",
            "status": "accounted",
            "bank_account_name": "taoli",
            "bank_account_no": "102280067984",
            "account_id": "acct_2WoHFSjAh1eaRpVH72RsOc1OyCw",
            "trans_at": "1697473288000",
            "object": "disburse_order"
        }
    ]
}
```


5.3 余额(balance)

5.3.1 查询余额

字段名	说明	类型
receivable_balance	商户应收款余额	string
cash_balance	商户可提现或可代付资金总额	string
In_transit_balance	商户在途资金总额	string

应答消息示例：

```
{
    "cash_balance": "53.38",
    "in_transit_balance": "30.00",
    "receivable_balance": "120.38"
}
```


6 回调通知(webhook)

不同的交易，在交易完成后，会回调通知商户。消息体为相应交易涉及的业务对象，比如代收的回调通知为 PaymentIntent 对象（5.1.1），代付的回调通知为 DisburseOrder 对象（5.2.1）。如果是交易失败了，可以在 fail_reason 得到失败原因。可以通过返回数据中的 object 字段区分对象类型进行不同的解析处理。

商户可以统一设置回调地址，或者在发送支付或代付请求时传 callback_url。

回调通知采用采用 API signing 的鉴权方式，在商户 onboarding 完成后，会分配给商户 accessSecret，不同的 accessSecret 对应相应的环境：测试环境/生产环境，类似：sk_test_xxxxx 或 sk_prd_xxxxxx。

签名方式：

payload: timestamp=${timestamp}&body=${body};

x-op-signature: 使用 accessSecret 对 payload 取 HmacSHA256 哈希值。

请求 body 的字符串规则：

content-type 为 application/json 时，取整个 json 字符串（目前只考虑 application/json 的情况）；

参与签名的字段使用的消息头：

x-op-timestamp:1525872629832

x-op-signature:xfX+bZxY2yl7EB/qdoDy9v/uscw3Nnj1pgoU+Bm6xdM=

6.1 示例

支付的回调消息类似如下：


```
{
    "id":"py_2WvNMWLz2joeCxs1VhxG0TVq02O",
    "ref_trade_id":"5000",
    "account_id":"acct_juancash",
```

```
    "amount":"10.00",
    "currency":"php",
    "payment_method":{
        "id":"",
        "type":"PAY_QRCODE",
        "billing_details":null,
        "card":null,
        "direct_debit":null,
        "bank_account":null,
        "metadata":null
    },
    "customer":null,
    "fee_amount":"0.38",
    "exchange_rate":"0",
    "status":"succeeded",
    "metadata":null,
    "fail_reason":"",
    "description":"AbC",
    "return_url":"",
    "callback_url":"",
    "trans_at":1697608780000,
    "object":"payment_order"
}
```


用于签名的 payload 为：


```
timestamp=1697609372293&body={"id":"py_2WvNMWLz2joeCxs1VhxG0TVq02O","ref_trade_id":"5000","account_id":"acct_juancash","amount":"10.00","currency":"php","payment_method":{"id":"","type":"PAY_QRCODE","billing_details":null,"card":null,"direct_debit":null,"bank_account":null,"metadata":null},"customer":null,"fee_amount":"0.38","exchange_rate":"0","status":"succeeded","metadata":null,"fail_reason":"","description":"AbC","return_url":"","callback_url":"","trans_at":1697608780000,"object":"payment_order"}
```


生成的签名为：


```
e1bad1563de6ab18fa4b6f6466dd9880a87186b6b45480a598c35630db9307af
```