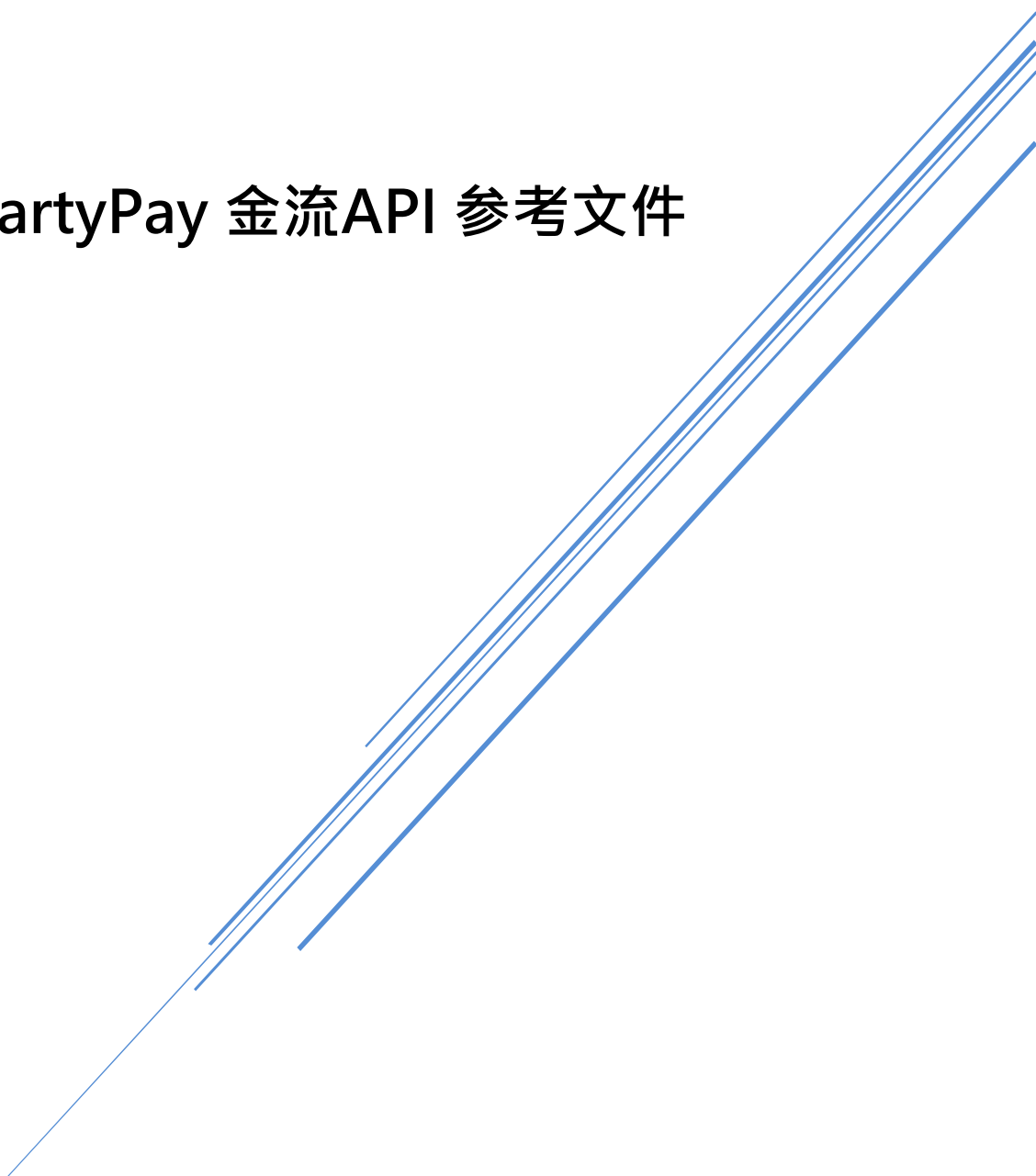


3rdPartyPay 金流API 参考文件



目录

版本纪录	2
1. 简介	3
2. 使用流程说明	4
2.1. 申请交易授权码	5
2.2. 让用户选择交易方式	5
2.3. 通知厂商交易结果	5
2.4. 查询交易纪录	5
3. 游戏API参考说明	5
3.1. 申请交易授权码	6
3.2. 将授权码传至3rd Party Pay开始交易	8
3.3. 通知厂商交易结果	10
3.4. 查询交易纪录	12
附录A	15
附录B	15
附录C	15

版本纪录

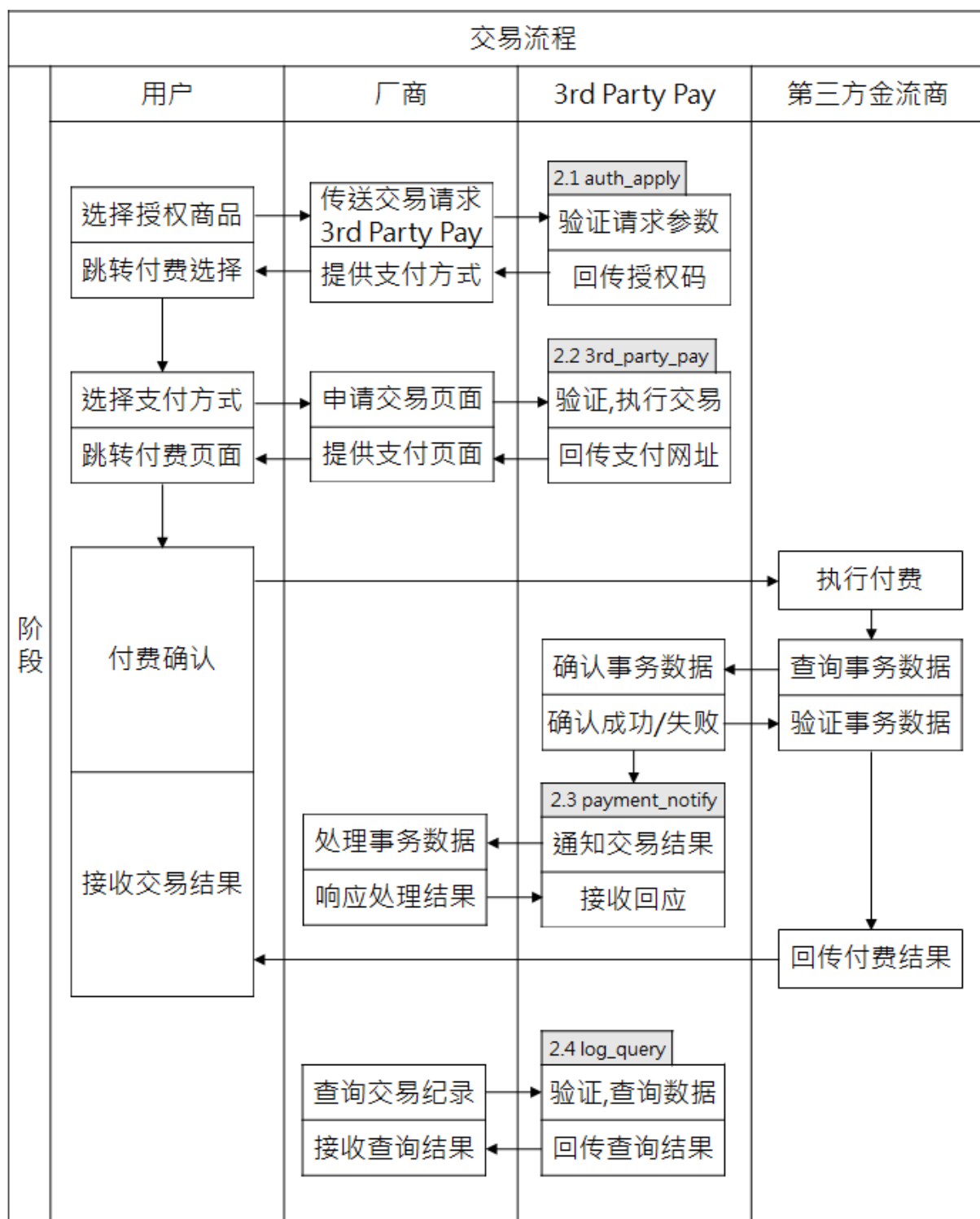
日期	版本	描述
2018-03-01	0.1.0	初版
2018-03-06	0.1.1	修改API网址及小写英文
2018-03-15	0.1.2	网址参数及加底线
2018-04-09	0.1.3	修改trade_query回传
2018-04-12	0.1.4	新增trade_log_query
2018-04-13	0.1.5	优化运作流程
2018-05-05	0.1.6	更新payment_notify及部分更新
2018-07-21	0.1.7	在通知接口加入驗證碼驗證
2018-08-06	0.1.8	修改通知接口回应需求
2018-08-16	0.1.9	增加authcode时效性
2018-08-23	0.2.0	新增支付宝扫码所支持金额
2018-10-25	0.2.1	调整厂商交易序号限制长度为8~30

1. 简介

提供下列API程序

程序名称	描述
auth_apply	申请交易授权码
3rd_party_pay	将授权码传至3rd Party Pay · 开始交易
payment_notify	3rd Party Pay通知交易结果
log_query	查询3rd Party Pay交易纪录

2. 使用流程说明



交易流程图

2.1. 申请交易授权码

厂商进行交易前，须先向3rd Party Pay申请交易授权码，并取得支付方式提供给用户。
呼叫API详情请见章节3.1。

2.2. 让用户选择交易方式

用户在选择交易方式，选择完毕厂商连同验证码传至3rd Party Pay开始进行交易，API
详情请见章节3.2。

2.3. 通知厂商交易结果

在3rd Party Pay与第三方金流确认交易后，会根据auth_apply填入的notify_url主动通知厂商交易结果，厂商成功接收并处理完资料后回应成功，API详情请见章节3.3。

2.4. 查询交易纪录

厂商可自行输入时间区间查询交易纪录，3rd Party Pay依据输入时间回传查询结果，
查询时间区间最长为24小时，API详情请见章节3.4。

3. 游戏API参考说明

测试帐号

测试厂商编号(company_service_id)：cdaecb46830eb3cb920d8b9856e6438e

测试厂商密钥(secret key)：97bc26ce95637a9114fce9d72bde884f

符号约定

符号	说明
M	必填
O	可选
C	条件填
S	字串

S(n)	长度为n的字串
N	数字

3.1. 申请交易授权码

厂商在进行相关交易之前，须先与3rd Party Pay取得相关本次交易之授权码作为本次交易之凭证，同时传递付费方式给用户，验证码具有时效性五分钟。参数中需请厂商设定交易结束后接收交易通知的网址。

请求URL

URL	https://api.166ok.com/api/v1.0/auth_apply
方法	GET

参数

字段名	参数名	型别	必填	说明
厂商服务代码	company_service_id	S(30)	M	由3rd Party Pay编列
厂商交易序号	trade_service_id	S	M	厂商自订，每笔不得重复，长度8~30
交易模式	trade_type	S(1)	M	0:non 1:web 2:android 3:ios
用户代号	customer_id	S(30)	O	用户于厂商的唯一识别编号
品项代码	item_code	S(15)	M	由厂商编列
品项名称	item_name	S(20)	M	由厂商编列
交易金额	amount	N	M	产品金额，限制为 $1 \sim 2^{32}$ ，小数点精细度至第六位

币别	currency	S(3)	M	币别代码列表参考 附录B 说明
跳转url	finish_url	S(128)	M	交易结束要用户跳转的网址
通知url	notify_url	S(128)	M	交易后通知厂商结果的网址
时间戳记	timestamp	S(20)	M	日期格式: "Y-m-d H:i:s"
验证码	token	S(32)	M	产生方式参考 附录C 说明，长度必須為32

请求格式范例

GET /auth_apply

```

company_service_id: companyserviceid0001
trade_service_id: tradeserviceid0001
trade_type: 2
customer_id: user001
item_code: product001
item_name: 3PP_游戏充值
amount: 150
currency: 1
finish_url: http://web_address/web/trade_finish
notify_url: http://web_address/api/trade_notify
timestamp: 2017 -02-01 00:00:00
token : 6B9E004862748B5080457B623DCCB181

```

回应格式范例(JSON)

```

{
  "return_code": 1,
  "return_msg": "申请成功",
  "auth_code": "7B5489E16FD2A173011AFC8153866FB0562351AA1",
  "payment_selection": {

```



```

    "银联快捷": 5,
    "QQ扫码": 6,
    "支付宝扫码": 20
  }
}

```

栏位说明

字段名	参数名	型别	必填	说明
授权结果代码	return_code	N(1)	M	1为成功；其他为失败 ◎详细原因可参return_msg
讯息描述	return_msg	S(128)	M	对于return_code描述
以下参数当授权成功，即return_code为1时，回传				
交易授权码	auth_code	S(64)	C	申请成立3rd Party Pay提供 验证码做后续服务所使用
付费方式	payment_selection	S(128)	C	准备提供给用户的付费方式， 以阵列表示

3.2. 将授权码传至3rd Party Pay开始交易

厂商取得交易授权码及付费方式后，将付费方式提供给用户做选择，用户即可进行交易，将用户选择的付费方式连同授权码通知3rd Party Pay，交易成立后产生交易序号(trade_seq)对应一笔交易。

请求URL

URL	https://api.166ok.com/api/v1.0/3rd_party_pay
方法	GET

参数

字段名	参数名	型别	必填	说明
交易授权码	auth_code	S(64)	M	于3.1所申请之交易授权码
付费方式	payment_type	S(3)	M	代码列表参考 附录A 说明，或3.1回传之通道

请求格式范例

GET /3rd_party_pay

auth_code: 7B5489E16FD2A173011AFC8153866FB0562351AA1

payment_type: 6

回应格式范例(JSON)

```
{
  "return_code": 1,
  "return_msg": "操作成功",
  "trade_seq": "rGCMG99K6XPOvGRhrhbPfCpvGN2Q3sye ",
  "trade_service_id": "tradeserviceId001",
  "payment_type": "QQ扫码",
  "customer_id": "user001",
  "amount": 150,
  "currency": "RMB",
  "qrcode_url": "http://3rdPartyPay/paytype01/qrcode_url",
  "timestamp ": "2017-02-01 00:00:00",
  "token ": "6B9E004862748B5080457B623DCCB181"
}
```

栏位说明

字段名	参数名	型别	必填	说明
授权结果代码	return_code	N(1)	M	1为成功；其他为失败

				◎详细原因可参return_msg
讯息描述	return_msg	S(128)	M	对于return_code描述
以下参数当查询成功，即return_code为1时，回传				
交易序号	trade_seq	S(32)	C	交易成立于3rd Party Pay所纪录之交易序号
厂商交易序号	trade_service_id	S(30)	C	回传厂商交易序号
付费方式	payment_type	S (15)	C	直接以付费方式的名稱回傳
会员代号	customer_id	S(30)	O	用户于厂商的唯一识别编号，申請時授权码有填才回應
金额	amount	N	C	交易金额
币别	currency	S(3)	C	直接以币别名称回传
QRcode连结	qrcode_url	S(128)	C	连结可直接生成QRcode
时间戳记	timestamp	S(20)	C	格式: Y-m-d H:m:s
验证码	token	S(64)	C	产生方式参考 附录C 说明

3.3. 通知厂商交易结果

3rd Party Pay会透过此介面主动以GET方式通知厂商交易结果是否完成，会根据厂商于3.1 auth_apply传来的notify_url通知，厂商接收成功须回传接收成功

参数

字段名	参数名	型别	必填	说明
交易状态	pay_state	N(1)	M	交易状态代码，详情请参考pay_summary 1: 未支付 2: 交易成功，未回调 3: 交易结束

				4 : 交易失败
交易状态描述	pay_summary	S(128)	M	交易详细描述
交易序号	trade_seq	S(32)	C	交易成立于3rd Party Pay所纪录之交易序号
厂商交易序号	trade_service_id	S(30)	M	回传厂商交易序号
用户代号	customer_id	S(30)	O	用户于厂商的唯一识别编号 · 申请时授权码有填才通知
付费方式	payment_type	S(15)	M	直接以付费方式的名稱通知
金额	amount	N	M	交易金额
币别	currency	S(3)	M	直接以币别名称回传
交易申请时间	pay_start_time	S(20)	M	日期格式: "Y-m-d H:i:s"
交易完成时间	pay_end_time	S(20)	M	日期格式: "Y-m-d H:i:s"
验证码	token	S(32)	M	参考 附录C 说明 · 長度為32

请求格式范例

GET /

```
"pay_state": 2,
"pay_summary": "交易成功",
"trade_seq": "rGCMG99K6XPOvGRhrhbPfCpvGN2Q3sye ",
"trade_service_id": "tradeserviceid0001",
"customer_id": "user001",
"payment_type": "QQ扫码",
"amount": 150,
"currency": "RMB",
"pay_start_time": "2017-02-01 00:00:00",
```

```
"pay_end_time": "2017-02-01 00:02:00",  
"token": "adbfa0a311df6c8fee15358482388567"
```

回应格式范例(text)

```
"success"
```

栏位说明

成功接收后，请核对交易金额、付费方式、验证码等资讯。确认资料无误请回复"success"则3rd Party Pay为成功；其他于回复3rd Party Pay皆认定为失败

3.4. 查询交易纪录

厂商透过此介面向3rd Party Pay根据时间进行交易查询。

请求URL

URL	https://api.166ok.com/api/v1.0/trade_query
方法	GET

参数

字段名	参数名	型别	必填	说明
厂商服务代码	company_service_id	S(30)	M	由3rd Party Pay编列之代码
查询起始时间	start_time	S(20)	M	日期格式: "Y-m-d H:i:s"
查询结束时间	end_time	S(20)	M	日期格式: "Y-m-d H:i:s" · 且不得超過當日時間

请求格式范例

```
GET/trade_query
```

company_service_id: companyserviceid0001
start_time : 2017-02-01 13:21:00
end_time : 2017-02-02 23:59:59

回应格式范例(JSON)

```
{
  "return_code": 1,
  "return_msg": "查询成功",
  "return_data": [
    {
      "pay_state": 3,
      "pay_summary": "交易结束",
      "trade_seq": "rGCMG99K6XPOvGRhrhbPfCpvGN2Q3sye",
      "trade_service_id": "tradeserviceid0001",
      "item_code": "product001",
      "item_name": "产品编号001",
      "payment_type": "QQ扫码",
      "amount": 150,
      "currency": "RMB",
      "pay_start_time": "2017-02-02 00:00:00",
      "pay_end_time": "2017-02-02 00:02:00 ",
    },
    {
      "pay_state": 1,
      "pay_summary": "交易中",
      "trade_seq": "z4tZL64VfGkLLwIDB2YodHJwDjQuLMUV",
      "trade_service_id": "tradeserviceid0001",
      "customer_id": "user001",
      "item_code": "product001",
      "item_name": "产品编号002",
      "payment_type": "QQ扫码",
      "amount": 300,
```

```

        "currency": "RMB",
        "pay_start_time": "2017-02-01 17:00:00",
    }
]
}

```

栏位说明

字段名	参数名	型别	必填	说明
授权结果代码	return_code	N(1)	M	1为成功；其他为失败 ◎详细原因可参考return_msg
讯息描述	return_msg	S(128)	M	对于return_code描述
以下参数当查询成功，即return_code为1时，回传				
return_data	3rd Party Pay回传交易纪录，为字串阵列型态，纪录包含			
	参数名	说明	必填	
	pay_state	交易状态代码，详情请参考pay_summary 1: 未支付 2: 交易成功，未回调 3: 交易结束 4: 交易失败	M	
	pay_summary	交易状态描述	M	
	trade_seq	交易申请成立于3rd Party Pay所纪录之交易序号	M	
	trade_service_id	厂商自订之交易序号	M	
	customer_id	用户代码，用户于厂商的唯一识别编号， 申请时授权码有填才回应	C	
	item_code	产品代码	M	

	payment_type	付费方式	M
	amount	金额名称	M
	currency	币别名称	M
	pay_start_time	交易申请时间，日期格式: "Y-m-d H:i:s"， 若訂單狀態(pay_state)為0，不回傳	C
	pay_end_time	交易完成时间，日期格式: "Y-m-d H:i:s"， 若訂單狀態(pay_state)為0或1，不回傳	C

附录A

付费方式代码列表

付费金流代码	说明
6	qq扫码
9	银联扫码
20	支付宝扫码
21	支付宝WAP

附录B

币别代码列表

币别代码	说明
1	CNY

附录C

API验证码产生方式

步骤1

将参数依文件顺序组成字符串

以3.1 auth_apply传至3rd Party Pay参数为例

encode_value =

company_service_id + trade_service_id + trade_type + item_code + item_name + amount + currency + timestamp + 厂商的secret key

以3.2 3rd_party_pay回传厂商的回应为例

encode_value =

return_code + trade_seq + trade_service_id + payment_type + amount + currency + timestamp + 厂商的sceret key

以3.3 通知厂商交易结果为例

encode_value =

pay_state + pay_summary + trade_seq + trade_service_id + payment_type + amount + currency + pay_start_time + pay_end_time + 厂商的secret key

步骤2

编码后的字符串用MD5函数作杂凑值当作验证码

token = MD5(encode_value)

范例

token = 6b9e004862748b5080457b623dccb181