

# GGC接口文档

版本：v1.3.2

阅读对象：技术人员、产品经理

审核人员：\*\*\*

## 钱包APP

---

- 下载地址：[待定]

## 域名

---

- 沙盒环境: <http://ggc-tradegate.ggcw.red:8084/v1>
- 正式环境: [待定]

## 通信协议

---

- 下述所有API接口均须通过WebService的方式进行调用；
- 所有传参均需要进行验签。
- 数据提交方式均通过application/x-www-form-urlencoded的方式POST提交。

参数配置：

- app\_id: 应用ID
- app\_key: 用户Key
- app\_secret: 应用密钥
- URL: 服务地址，如<http://ggc-tradegate.ggcw.red:8084/v1/>

备注：请联系平台方索要上述参数的具体值。

## API列表

---

序号	类别	接口	功能说明	备注
1	商户	<a href="#">/mch/take</a>	通用支付	支付授权方式
2	商户	<a href="#">ggcpay:pay</a>	APP唤起式支付	Url Schema调用方式。将该url转换成二维码，亦即是扫码支付。
3	商户	<a href="#">/mch/send</a>	给玩家退款	也可用于游戏提现、挖矿奖励
4	商户	<a href="#">/mch/txdetail</a>	交易详情	供事后核查详情
5	商户	<a href="#">结果回调</a>	交易通知	交易完成后，由GGC清算节点网关发给商户服务器

备注：

- 可以在后台进行商户钱包地址的绑定
- 如有发生异常时请参考 [错误码说明](#)

## 1.通用支付-支付授权方式（mch/take）

简要描述：

- 功能：获得交易的授权码，需要用户使用该授权码进行交易授权。
- 流程：①（商户APP）发起扣费申请->②（商户APP）获得动态授权码->③（商户APP）向用户展示动态授权码->④（商户APP）用户复制动态授权码->⑤【钱包APP】用户进入相应钱包->⑥【钱包APP】用户打开授权功能,粘贴该动态授权码,确认授权->⑦【GGC服务器】推送结果给（商户服务器）->⑧（商户服务器）更新扣费状态。
- 适用场景：适用于网站、微信小程序等，无法直接唤起钱包APP、无法使用扫码支付的应用环境。

请求URL：

- URL + `/mch/take?app_id=应用ID&app_key=应用KEY`

请求方式：

- POST application/x-www-form-urlencoded

请求参数：

参数名	必选	类型	说明
amount	是	Number	ggc数量
goods	是	String	商品名，原样返回
method	否	String	Http Method，原样返回
user_data	否	String	用户自定义数据，可以是订单号等，原样返回
notify_url	是	String	接收结果通知的服务地址
sign	是	String	MD5签名,注：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值

MD5签名

sign的值为：md5((amount=参数值&goods=参数值&method=参数值&notify\_url=参数值&user\_data=参数值)+应用密钥值)

注意：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值！！

请求示例

```
POST http://ggc-tradegate.ggcw.red:8084/v1/mch/take?app_id=iuwwoq1&app_key=1bdc23343 HTTP/1.1
Content-Type: application/x-www-form-urlencoded;charset=utf-8

amount=123.00&goods=%E9%99%90%E9%87%8F%E7%A4%BC%E5%8C%85&method=POST&user_data=10291023123&notify_url=http%3A%2F%2Fyour-server.com%2Fnotify%2Fhandler&sign=134389a8sdfa832423asdss686868asdfasd7899a9sdaas8797asaa87tsx
```

响应参数：

参数名	类型	说明
txid	String	交易号
auth_code	String	动态授权码，商户需要将该授权码展示给用户看，用户将根据此码进行授权

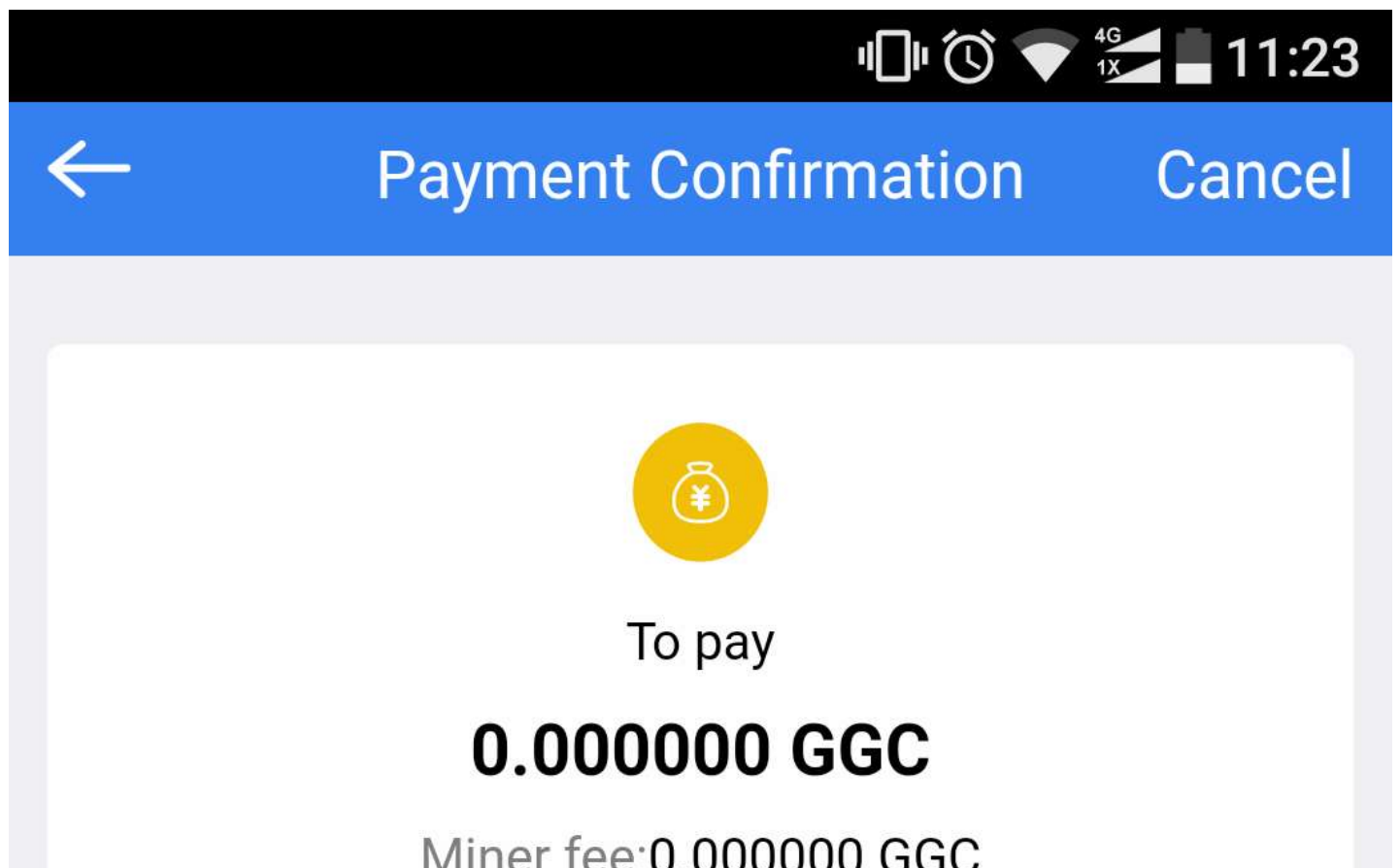
返回示例

```
{
  "result": {
    "txid": "8c3798ded3e3d83f4ab15cb7f86c293201906cca4b5ca997063e1ff3e0e39115"
  },
  "auth_code": "A23B23",
},
"code": 0,
"msg": ''
}
```

## 2.APP唤起式支付-Url Schema方式 (ggcpay:pay)

### 简要描述：

- 调起钱包APP进行支付，或生成支付二维码，异步 返回支付结果。
- 功能：本操作完成GGC从玩家钱包地址到商家绑定钱包地址的转移。
- 流程：① (商户APP) 采集用户的消费信息-> ② (商户APP) 通过UrlSchema唤起【钱包APP】，并传递参数-> ③【钱包APP】请求【GGC服务器】生成支付信息-> ④【钱包APP】用户点确认支付-> ⑤【GGC服务器】将支付结果推送给 (商户服务器) -> ⑥ (商户服务器) 进行通知处理后返回ok。
- 适用场景：i. 适合手游、移动商城等APP，可以直接通过UrlSchema唤起钱包APP进行订单支付。ii. 或用于端游、无人售货机等，可以通过扫描二维码进行订单支付。
- 示例图：



---

Merchant: International Community  
Development

Goods: test goods

Book balance:0 GGC

Spendable balance:0 GGC

---

Submit

Hint:After the payment is confirmed, the actual deduction will be made from your account **Amount to be paid** and **Miner fee**

请求URL:

- `ggcpay:pay?app_id=应用ID&app_key=应用KEY`

请求方式:

- Url Schema

请求参数：

参数名	必选	类型	说明
amount	是	Number	ggc数量
goods	是	String	商品名，原样返回
method	否	String	Http Method，原样返回
user_data	否	String	用户自定义数据，可以是订单号等，原样返回
notify_url	是	String	接收结果通知的服务地址
sign	是	String	MD5签名，注：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值

MD5签名

sign的值为：md5((amount=参数值&goods=参数值&method=参数值&notify\_url=参数值&user\_data=参数值)+应用密钥值)

注意：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值！！

请求示例

```
var amount="29.90", user_data="234212344433", goods="限量礼包", method="POST", notify_url="http://your-server.com/notify/handler";

var sign = md5("amount="+amount+"&goods="+ goods+"&method="+ method+"&notify_url="+notify_url+"&user_data="+user_data+app_secret);

var url_schema = "ggcpay:pay?app_id=idreamsky&app_key=ggc&amount="+encodeURIComponent(amount)+"& goods="+encodeURIComponent(goods)+"&method="+encodeURIComponent(method)+"&notify_url="+encodeURIComponent(notify_url)+"&user_data="+encodeURIComponent(user_data)+"&sign="+encodeURIComponent(sign);

//唤起
window.open(url_schema);

//或生成支付二维码
your_method_convertStringToQrcode(url_schema);
```

响应参数：

无

返回示例：

无

### 3.给玩家退款（mch/send）

简要描述：

- 功能：向用户转出，可用于退款、用户奖励、挖矿等， 异步 返回支付结果
- 流程：① （商户服务器） 向【GGC服务器】发起向用户转账的请求-> ② 【GGC服务器】完成处理-> ③ 【GGC服务】将结果推送给 （商户服务器） -> ④ （商户服务器） 处理通知并在完成后返回ok字符串。
- 适用场景：如退款、给用户发放奖励等。

请求URL：

- URL + /mch/send?app\_id=应用ID&app\_key=应用KEY

请求方式：

- POST application/x-www-form-urlencoded

请求参数：

参数名	必选	类型	说明
amount	是	Number	ggc数量
goods	是	String	商品名，原样返回
method	否	String	Http Method，原样返回
user_data	否	String	用户自定义数据，可以是订单号等，原样返回
to_address	是	String	接收钱包地址
notify_url	是	String	接收结果通知的服务地址
sign	是	String	MD5签名，注：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值

#### MD5签名

sign的值为：md5((amount=参数值&goods=参数值&method=参数值&notify\_url=参数值&to\_address=参数值&user\_data=参数值)+应用密钥值)

注意：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值！！

请求示例

```
POST http://ggc-tradegate.ggcw.red:8084/v1/mch/send?app_id=iuwwoq1&app_key=1bdc23343 HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8

amount=123.00&goods=%E9%99%90%E9%87%8F%E7%A4%BC%E5%8C%85&method=POST&user_data=10291023123&to_address=Gdeqws8s8d89adsxuyyasdt5312&notify_url=http%3A%2F%2Fmerchant-server.com%2Fnotify%2Fhandler&sign=134389a8sdfa832423asdss686868asdfasd7899a9sdaas8797asaa87tsx
```

响应参数：

参数名	类型	说明
txid	String	交易号

返回示例

```
{
  "result": {
    "txid": "dc30a0b404f08cf1b45cd43e6a1ea9436c02f30d2d2005456ef9272ba912b208"
  },
  "code": 0,
  "msg": ''
}
```

## 4.交易详情（mch/txdetail）

简要描述：

- 查询一笔交易明细

请求URL：

- URL + /mch/txdetail?app\_id=应用ID&app\_key=应用KEY

请求方式：

- POST application/x-www-form-urlencoded

请求参数：



参数名	必选	类型	说明
txid	是	String	交易号
sign	是	String	MD5签名，注：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值

## MD5签名

sign的值为：md5((txid=参数值)+应用密钥值)

注意：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值！！

## 请求示例

```
POST http://ggc-tradegate.ggcw.red:8084/v1/mch/txdetail?app_id=iuwwoq1&app_key=1bd
c23343 HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8

txid=dc30a0b404f08cf1b45cd43e6a1ea9436c02f30d2d2005456ef9272ba912b208&sign=ouweoia
sdf1kaslw323242u3l234kas
```

## 响应参数：

参数名	类型	说明
txid	String	交易号，唯一
time	String	交易时间，Unix时间戳
trade_type	String	业务类型，pay：订单支付，send：支付给玩家
from_address	String	支出方钱包地址
to_address	String	收入方钱包地址
goods	String	商品名，原样返回
method	String	Http Method，原样返回
user_data	String	用户自定义数据，可以是订单号等，原样返回
items	Object	交易明细

## 交易明细 (items)：

参数名	类型	说明
amount	Number	ggc数量
fee	Number	矿工手续费，由支出方承担

#### 返回示例

```
{
  "result": {
    "txid": "a8dea63045a0faad558d63156fe3266287c31447092fd4692f50a27faf73ba4a",
    "time": 1531379847,
    "trade_type": "send",
    "from_address": "GPi8mtGFgspHpXDQcgCQ9n4Sg5XrPQqw7p",
    "to_address": "GC56yuPO45oiPXDQcgCQ9n4Sg5XrPQ0oprw",
    "goods": "限量礼包",
    "method": "POST",
    "user_data": "wers23rasd",
    "items": [
      {
        "amount": 1,
        "fee": 0.000052
      }
    ]
  },
  "code": 0,
  "msg": ""
}
```

## 5.交易结果通知（notify\_url）

#### 简要描述：

- 平台将交易结果通知发给商户应用进行处理
- 商户收到通知并处理完毕后返回ok字符串给平台
- 适用场景：接收如订单支付、发送、扣费、充值、提现等的异步结果返回。

#### 通知URL：

- 在存入/取出接口中设置的 `{notify_url}` 参数

#### 通知方式：

- POST application/x-www-form-urlencoded

通知参数：

参数名	类型	说明
data	String	数据,json字符串
sign	是	String

MD5签名

sign的值为：md5((data=参数值)+应用密钥值)

注意：参与计算签名的参数值，一定是原始值，不能是urlencode()之后的值！！

数据（data）：

参数名	类型	说明
txid	String	交易号，唯一
time	String	交易时间，Unix时间戳
trade_type	String	业务类型，pay：订单支付，send：支付给玩家
from_address	String	支出方钱包地址
to_address	String	收入方钱包地址
goods	String	商品名，原样返回
method	String	Http Method，原样返回
user_data	String	用户自定义数据，原样返回
items	Object	交易明细

交易明细（items）：

参数名	类型	说明
amount	Number	ggc数量
fee	Number	矿工手续费，由支出方承担

返回示例

\$\_POST['data']的数据示例：

```
{
  "result": {
    "txid": "o5qknixhsdxm0lyebzyzw6wdb4be8qtxmn2zivqw8ghvar6eb0akdrbhexddite3"
  },
  "time": 1538207496,
  "trade_type": "pay",
  "from_address": "GZC3h7iaf5vx7SQJafwJXWAgHQUPFwbTbF",
  "to_address": "GP3Dk6x4HDvFgF323FrX76FgXXj2YGRfnw",
  "goods": "\u7535\u89c6\u5361\u70b9",
  "method": "POST",
  "user_data": "10010122341",
  "items": {
    "amount": "0.000000",
    "fee": "0.000000"
  }
},
"code": 0,
"msg": ""
}
```

\$\_POST['sign']的数据示例:

bef8f60dc7dcd2e930023f3c5cc1bb55

商户应用处理

商户的回调地址上接到GGC清算节点网关推送过来的交易结果，并进行相关的自有业务逻辑处理之后，在业务逻辑处理成功后返回 `ok` 字符串。

如果没有返回或返回的是其他值，则GGC清算节点网关认为商户处理失败。

商户应用处理结果示例

ok

注意：该值必须返回！

错误码说明

代码	说明
0	成功
1	读写数据库失败

2	短信操作达到上限，请明天再试
3	短信操作达到次数上限
4	短信尝试过于频繁
5	短信操作过于频繁
6	未知手机号
7	无效的短信验证码
8	尚未激活
9	此账号已被冻结
10	短信发送失败
11	该短信验证码已过期
12	区块链节点连接失败
13	缺少应用接入参数
14	应用接入认证失败
15	未知的商户
16	发送方小钱包无效
17	接收方小钱包无效
18	无效的商户绑定钱包
19	无效的玩家钱包
20	区块链操作失败
21	数据库写入失败
22	权限不足
23	没有匹配的交易号
24	超出权限范围
25	无效的动态授权码
26	钱包余额不足以支付矿工手续费
27	钱包余额不足以支付

28	试图提现到小钱包
29	发送失败
30	无效的数据签名
31	不允许清算
32	无效的矿池
33	该相应钱包不属于此商户
34	无效的开发者的账号
35	无效的点卡
36	该相应钱包和发卡单位不匹配
37	过期的点卡
38	无效的金额
39	不同类型的相应钱包
40	未开通相应相应钱包
41	无效的交易
42	未设置清算对家
43	清算对家钱包不可用
44	授权码已过期
45	服务商储备金不足
46	无效的接收账号
47	未激活该相应钱包
48	无效的操作请求
49	手机号码被占用
50	发卡商钱包异常