

BITGAMEX 交易所和游戏中心交互文档

目录

1. 版本日志	3
2. 游戏中心提供的接口	4
2.1. 交易所转账给游戏中心接口	4
2.1.1. 接口说明	4
2.1.2. 接口参数	4
2.1.3. 响应结果	5
2.1.4. 特殊说明	6
3. 交易所提供的接口	7
3.1. 游戏中心转账给交易所接口	7
3.1.1. 接口说明	7
3.1.2. 接口参数	7
3.1.3. 响应结果	8
3.1.4. 特殊说明	8
3.2. 游戏中心转账给用户钱包接口	9
3.2.1. 接口说明	9
3.2.2. 接口参数	9
3.2.3. 响应结果	10
3.2.4. 特殊说明	11

1. 版本日志

修改记录

版本号	修改日期	变更说明	修改人	审核	备注
0.1	2018-3-5	起草	谷力		

2. 游戏中心提供的接口

2.1. 交易所转账给游戏中心接口

2.1.1. 接口说明

使用场景：交易所请求游戏中心服务器给某玩家转账，从交易所的账户里划拨一定的游戏币到游戏中心里。

接口使用者：交易所以 HTTPS GET 方式请求。

请求地址：[https://待定游戏中心内网 ip/?a=transfer_coin_to_game](https://待定游戏中心内网ip/?a=transfer_coin_to_game)

2.1.2. 接口参数

参数	字段类型	必填	说明
param_data	String	是	参数通过RSA加密后再通过base64编码后得到的字符串
sign	String	是	参数通过RSA加密后的签名

param_data 加密前的参数表：

参数	字段类型	必填	说明
transaction_id	String(200)	是	本次转账在交易所系统中的转账单号
exchange_accid	String(100)	是	转账请求方的交易所账号id
game_uid	Int(11)	是	游戏中心用户id
game_id	Int(11)	是	游戏中心游戏id
token_symbol	String(20)	是	币种：BGX, BTC, ETH, ...
amount	Double	是	转账的游戏币数量
time	Int(20)	是	时间戳

2.1.3. 响应结果

类型：json 结构

成功：

```
{
  "succ":1,
  "balance":Double, 玩家在游戏中心的游戏币余额
}
```

失败：

```
{
  "succ":0,
  "req":"transfer_coin_to_game",
  "errno":错误编号
  "errmsg":错误描述
}
```

errno值	说明
-999	ip受限
-998	参数不全
-997	参数错误
-996	校验失败
-995	协议不支持
-994	异常抛出
-993	游戏未开放
-992	读缓存失败
-991	不存在rpc服务器
-990	金币不足
-989	请求失败
-988	请求超时
-987	未确定的错误
-986	无效的钱包地址
-985	无效的交易所账号

* 以上为公共错误号定义，也为其他接口共享，后面不再重复列出。各接口特定的错误号会重复定义，结合错误返回结果中的 `req` 字段可唯一确定。

errno值	说明

2.1.4. 特殊说明

transaction_id: 本次转账在交易所系统中的转账单号，用于防止重复请求，具体逻辑由开发者实现。

验证签名以及解码参数: 为了确保通信安全，调用会使用交易所的私钥生成参数的签名，并使用游戏中心提供的公钥加密参数，签名和加密均使用RSA的签名和加密算法；

RSA签名验证及解密算法:

public_key: 交易所公钥，由交易所部门提供，用于验证参数签名使用；

private_key: 通过签名算法生成的私钥，由游戏中心自行保管，用于密文解密；

param_data: 接口接收到的参数，加密后的参数列表，base64编码的字符串；

sign: 参数签名

过程如下：

1. 将获得的param_data通过base64解码，得到字节数组param_byte；
2. 游戏中心使用自己持有的私钥对参数解密，得到解密后的参数param（如果无法解密，尝试128字节分段解密）；
3. 使用交易所提供的公钥对param进行签名校验，确保调用是由交易所发起的；
4. 对param进行字符串解析，交易所对参数中的特殊字符已进行url编码，游戏中心可使用解析url的方式对param进行解析（有些服务器会自动解码，请自行确认）；
5. 根据得到的参数进行逻辑处理；
6. 返回上述响应结果，结束本次回调（请确保对验证通过的请求回应"succ":1）。

注：以上操作涉及的字符都是以utf-8编码。

接口调用成功后，交易所将锁定的游戏币减记。

3. 交易所提供的接口

3.1. 游戏中心转账给交易所接口

3.1.1. 接口说明

使用场景：游戏中心请求交易所服务器给某玩家转账，从游戏中心的账户里划拨一定的游戏币到交易所里。

接口使用者：游戏中心以 HTTPS GET 方式请求。

请求地址：[https://待定交易所内网 ip/?a=transfer_coin_to_exchange](https://待定交易所内网ip/?a=transfer_coin_to_exchange)

3.1.2. 接口参数

参数	字段类型	必填	说明
param_data	String	是	参数通过RSA加密后再通过base64编码后得到的字符串
sign	String	是	参数通过RSA加密后的签名

param_data 加密前的参数表：

参数	字段类型	必填	说明
transaction_id	String(200)	是	本次转账在游戏中心系统中的转账单号
game_uid	Int(11)	是	转账请求方的游戏中心用户id
exchange_accid	String(100)	是	交易所账号id
token_symbol	String(20)	是	币种：BGX, BTC, ETH, ...
amount	Double	是	转账的游戏币数量
time	Int(20)	是	时间戳

3.1.3. 响应结果

类型: json 结构

成功: (如果该笔转账在交易所里之前已标记为成功, 也请务必回复成功 succ=1)

```
{
  "succ":1,
  "balance":Double, 玩家在交易所的游戏币余额
}
```

失败:

```
{
  "succ":0,
  "req":"transfer_coin_to_exchange",
  "errno":错误编号
  "errmsg":错误描述
}
```

errno值	说明

3.1.4. 特殊说明

transaction_id: 本次转账在游戏中心系统中的转账单号, 用于防止重复请求, 具体逻辑由开发者实现。

验证签名以及解码参数: 为了确保通信安全, 调用会使用游戏中心的私钥生成参数的签名, 并使用交易所提供的公钥加密参数, 签名和加密均使用RSA的签名和加密算法;

RSA签名验证及解密算法:

public_key: 游戏中心公钥, 由游戏中心部门提供, 用于验证参数签名使用;

private_key: 通过签名算法生成的私钥, 由交易所自行保管, 用于密文解密;

param_data: 接口接收到的参数, 加密后的参数列表, base64编码的字符串;

sign: 参数签名

过程如下:

1. 将获得的param_data通过base64解码, 得到字节数组param_byte;
2. 交易所使用自己持有的私钥对参数解密, 得到解密后的参数param (如果无法解密, 尝试128字节分段解密);
3. 使用游戏中心提供的公钥对param进行签名校验, 确保调用是由游戏中心发起的;
4. 对param进行字符串解析, 游戏中心对参数中的特殊字符已进行url编码, 交易所可使用解析url的方式对param进行解析 (有些服务器会自动解码, 请自行确认);
5. 根据得到的参数进行逻辑处理;
6. 返回上述响应结果, 结束本次回调(请确保对验证通过的请求回应"succ":1)。

注: 以上操作涉及的字符都是以 **utf-8** 编码。

接口调用成功后, 游戏中心将锁定的游戏币减记。

3.2. 游戏中心转账给用户钱包接口

3.2.1. 接口说明

使用场景: 游戏中心请求交易所服务器给某玩家的钱包转账, 从游戏中心的账户里划拨一定的游戏币到玩家指定的钱包 (经由交易所) 里。

接口使用者: 游戏中心以 HTTPS GET 方式请求。

请求地址: https://待定交易所内网 ip/?a=transfer_coin_to_wallet

3.2.2. 接口参数

参数	字段类型	必填	说明
param_data	String	是	参数通过RSA加密后再通过base64编码后得到的字符串
sign	String	是	参数通过RSA加密后的签名

param_data 加密前的参数表:

参数	字段类型	必填	说明
transaction_id	String(200)	是	本次转账在游戏中心系统中的转账单号
game_uid	Int(11)	是	转账请求方的游戏中心用户id
exchange_accid	String(100)	是	交易所账号id
wallet_address	String(50)	是	钱包地址
token_symbol	String(20)	是	币种: BGX, BTC, ETH, ...
amount	Double	是	转账的游戏币数量
time	Int(20)	是	时间戳

3.2.3. 响应结果

类型: json 结构

成功: (如果该笔转账在交易所里之前已标记为成功, 也请务必回复成功 succ=1)

```
{
  "succ":1,
  "balance":Double, 玩家在交易所的游戏币余额
}
```

失败:

```
{
  "succ":0,
  "req":"transfer_coin_to_wallet",
  "errno":错误编号
  "errmsg":错误描述
}
```

errno值	说明

3.2.4. 特殊说明

游戏中心转账到用户钱包，流程分两步走：

1. 游戏中心转账到指定交易所账户：这一步同 3.1 节所述；
2. 指定交易所账户再转账到用户个人钱包