

# 对接文档 v1.0

## 目录

0.API 列表	2
1.API 通讯与加密	2
1.1 API 格式	2
1.2 哈希计算	3
2.游戏启动接口	4
2.1 获取游戏链接	5
2.2 验证玩家身份	6
3.单一钱包接口	7
3.1 获取余额	7
3.2 玩家投注	8
4.转账钱包接口	10
4.1 获取游戏登录链接	10
4.2 转入额度	10
4.3 转出额度	11
4.4 查询玩家余额	13
4.5 查询单笔转账	14
5.通用接口	15
5.1 获取游戏列表	15
5.2 获取游戏记录	17
5.3 设置玩家返奖率	18
5.4 查询玩家返奖率	19
5.5 创建玩家	21
5.6 踢出玩家	21
6.附录	22
6.1 错误码	22
6.2 货币	23
6.3 语言	25
6.4 返奖率	26
6.5 须知	26

## 0.API 列表

接口	说明	备注
/getGameURL	获取游戏链接	游戏启动
/verify	玩家身份认证 (运营商提供)	游戏启动
/balance	获取玩家余额 (运营商提供)	单一钱包
/bet	玩家投付请求 (运营商提供)	单一钱包
/loginGame	获取游戏登录链接	转账钱包
/transferIn	为玩家转入额度	转账钱包
/transferOut	为玩家转出额度	转账钱包
/getBalance	查询玩家余额	转账钱包
/getTransaction	查询单笔转账	转账钱包
/getGameList	获取游戏列表	通用接口
/getGameHistory	获取游戏记录	通用接口
/player/setRTP	设置玩家 RTP	通用接口
/player/getRTP	查询玩家 RTP	通用接口
/player/create	创建玩家	通用接口
/player/kick	踢出玩家	通用接口

## 1.API 通讯与加密

### 1.1 API 格式

#### API 格式

##### 请求格式:

API 使用 JSON 格式传输, 平台将在收到请求后检验内容格式。请在 HTTP 头加上:

**content-type: application/json**

支持 gzip1.1 压缩, HTTP 请求头: Accept-Encoding: gzip

##### 返回格式:

API 将返回 JSON 格式数据, 对于成功是和失败的 API 请求, 都会返回 HTTP 状态码为 200 的响应结果。

API 返回的内容格式如下:

**content-type: application/json**

### 示例：

成功返回

```
{
  "data": {
    [ API 返回的数据, 每个接口各不相同 ]
  },
  "error": 0,
  "message": "Success"
}
```

错误返回

```
{
  "data": null,
  "error": [ 错误代码 ],
  "message": "[ 错误信息 ]"
}
```

## 1.2 哈希计算

**哈希代码**通过以下公式来计算：

从请求 POST 参数获取**所有**参数（预期哈希）并附加到字符串：

- 1) 按字母表顺序对所有参数排序.
- 2) 移除值为空字符串的参数，即 value=""
- 3) 在 key1=value1&key2=value2 中附加它们.
- 4) 附加密钥，即：key1=value1&key2=value2&secret=SECRET&salt=SALT.
- 5) 使用 MD5 计算哈希.
- 6) 与哈希参数对比。如哈希错误，运营商需返回固定错误码 1004

### 示例（以获取游戏地址接口为例）

商户信息为：

OperatorId: abc123

SecretKey: 2b5b996469686a8966c1205318811f08

Salt: sfnbDdWvsRSHJvRdpiilKfxjZ8ZWRe3i

请求参数：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "operatorId": "abc123",
  "token": "user_123_token",
  "gameId": 31010,
  "language": "en"
}
```

排序后的请求参数:

```
{
  "gameId": 31010,
  "language": "en"
  "operatorId": "abc123",
  "token": " e89045458a990e04d684418a2b8a2228",
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
}
```

组合后的字符串为:

```
gameId=31010&language=en&operatorId=abc123&token=e89045458a990e04
d684418a2b8a2228&traceId=b3f37e57-2873-40b1aa95-f126c25ed311
```

附加密钥后:

```
gameId=31010&language=en&operatorId=abc123&token=e89045458a990e04
d684418a2b8a2228&traceId=b3f37e57-2873-40b1aa95-
f126c25ed311&secret=2b5b996469686a8966c1205318811f08&salt=sfnbDdW
vsRSHJvRdpiilKfxjZ8ZWRe3i
```

使用 MD5 计算后为:

```
hash: fbf68f91e250e4894e0f11923d3fc246
```

最后请求参数为:

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "fbf68f91e250e4894e0f11923d3fc246",
  "operatorId": "abc123",
  "token": "e89045458a990e04d684418a2b8a2228",
  "gameId": 31010,
  "language": "en"
}
```

## 2.游戏启动接口

通用的游戏启动流程: {2.1 获取游戏链接}->{2.2 验证玩家身份}, 此流程可自动创建玩家  
转账钱包的可选流程: {5.5 创建玩家}->{4.1 游戏登录}, 此流程必须先创建玩家

## 2.1 获取游戏链接

**{ APIDomain }/gi/v2/getGameURL**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	哈希值，长度为 32 位
operatorId	string	是	商户 ID
token	string	是	玩家令牌，长度限制 1 到 200 个字符
gameId	integer	是	游戏 ID
language	string	否	游戏显示语言
<b>Response – Object</b>			
gameUrl	string	https://	

错误码	描述
0	成功
1005	参数错误

\*语言：可以不传，则默认英文；如果传了，就需要加入到 hash 校验中

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "token": "user_123_token",
  "gameId": "31010",
  "language": "en"
}
```

### 返回示例

```
{
  "data": {
    "gameUrl": "http://apidomain-game/index.html"
  },
  "error": 0,
  "description": "Success",
}
```

## 2.2 验证玩家身份

**{ OperatorAPIDomain }/verify (运营商提供)**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	hash 值，长度为 32 位
operatorId	string	是	商户 ID
token	string	是	玩家令牌，长度限制 1 到 200 个字符
gameId	integer	是	游戏 ID
<b>Response – Object</b>			
playerId	string	运营商保证唯一 (3-32 位, 字母,数字,@,-,_)	
currency	string	货币代码	

错误码	描述
0	成功
1005	参数错误
1007	玩家令牌已过期

\*已创建的玩家不能变更币种（需校验币种）

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "token": "user_123_token",
  "gameId": "31010"
}
```

### 返回示例

```
{
  "data": {
    "playerId": "user001_USD",
    "currency": "USD",
  },
  "error": 0,
  "description": "Success",
}
```

### 3.单一钱包接口

#### 3.1 获取余额

**{ OperatorAPIDomain }/balance (运营商提供)**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	hash 值，长度为 32 位
operatorId	string	是	商户 ID
token	string	是	玩家令牌，长度限制 1 到 200 个字符
playerId	string	是	玩家 ID
<b>Response – Object</b>			
playerId	string	货币代码	
currency	string	货币代码	
balance	(Decimal,2)	玩家余额(Decimal,2)，最大位数限制为 9999999999	

错误码	描述
0	成功
1006	无效的玩家令牌
1007	玩家令牌已过期

\* balance 必须  $\geq 0$ ；只支持最多 2 个小数位，多余的小数位将被截断；**最大位数**也要限制

#### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "token": "user_123_token",
  "playerId": "abcd123_usd"
}
```

#### 返回示例

```
{
  "data": {
    "playerId": "user001_USD",
    "currency": "USD",
  }
}
```

```

    "balance": 1006,
  },
  "error": 0,
  "description": "Success",
}

```

## 3.2 玩家投注

### 注意：

- ✓ **transactionId** 是每笔交易的唯一识别码。一次投注只会创建一个 transactionId
- ✓ 调用为幂等，对于重试，应返回玩家当前的实际余额

### { OperatorAPIDomain }/bet (运营商提供)

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	hash 值，长度为 32 位
operatorId	string	是	运营商 ID
transactionId	string	是	交易的 <b>唯一</b> 标识符
token	string	是	玩家令牌
playerId	string	是	玩家 ID
currency	string	是	币种
gameId	integer	是	游戏 ID
roundId	string	是	注单 ID ( <b>唯一</b> )
parentRoundId	string	是	母注单 ID
betAmount	Decimal	是	投注额度
winAmount	Decimal	是	派彩额度
transferAmount	Decimal	是	玩家的输赢 ( <b>正负值</b> )
rtp	integer	是	本回合的 rtp 档位
isBuy	int	是	是否购买模式(0 否, 1 是)
isFeature	int	是	是否特殊旋转(0 否, 1 是)
isEnd	int	是	是否最后回合(0 否, 1 是)
isExtra	int	是	是否加注模式(0 否, 1 是)
settleTime	integer	是	结算时间
<b>Response – Object</b>			
playerId	string	玩家 ID	
currency	string	货币代码	
balance	(Decimal,2)	玩家余额(Decimal,2) ， 最大位数限制为 9,999,999,999	



错误码	描述
0	成功
1011	玩家余额不足
1012	玩家不存在
1013	投注异常

\*确认请求前，运营商需**先判断**：玩家当前余额  $\geq$  betAmount；否则，应返回“余额不足”

\*更新后余额 = 玩家当前余额 + transferAmount（正负值）

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abb56e057f20f883e",
  "operatorId": "abc123",
  "token": "user_123_token",
  "playerId": "abcd123_usd",
  "currency": "USD",
  "gameId": 31021,
  "roundId": "1727255871000000001",
  "parentRoundId": "1727255871000000001",
  "betAmount": 10,
  "winAmount": 108,
  "transferAmount": 98,
  "rtp": 0,
  "isBuy": false,
  "isFeature": false,
  "isEnd": false,
  "isExtra": false,
  "settleTime": 1727255876
}
```

### 返回示例

```
{
  "data": {
    "playerId": "user001_USD",
    "currency": "USD",
    "balance": 1006,
  },
  "error": 0,
  "description": "Success",
}
```

## 4.转账钱包接口

### 4.1 获取游戏登录链接

{ APIDomain }/gi/v2/loginGame

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	hash 值，长度为 32 位
operatorId	string	是	商户 ID
playerId	string	是	玩家 ID
gameId	integer	是	游戏 ID
language	string	否	游戏显示语言
Response – Object			
gameUrl	string	游戏链接	

错误码	描述
0	成功
1012	玩家不存在

- 转账钱包可使用该接口获取游戏地址并登录，需保证玩家账号已被正确创建
- 也可按{2.1 获取游戏链接}，{2.2 验证玩家身份}流程登录游戏（自动创建玩家账户）

### 4.2 转入额度

注意：

- ✓ transactionId 是每笔交易的唯一识别码。运营商需要为每笔转账请求发送新的 transactionId。
- ✓ 幂等\_对于重复请求，我方不对玩家钱包余额进行任何操作，但返回转账成功的响应。

{ APIDomain }/gi/v2/transferIn

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符 请使用 UUID
hash	string	是	hash 值，长度为 32 位
operatorId	string	是	商户 ID
transactionId	string	是	交易的唯一标识符
playerId	string	是	玩家 ID

amount	Decimal(10,2)	是	转入额度
<b>Response – Object</b>			
transactionId	string	交易的 <b>唯一</b> 标识符	
currency	string	货币代码	
amount	Decimal(10,2)	本次转入额度（最大位数限制）	
balance	Decimal(10,2)	转账完成后的玩家余额（无论成功或失败）	

错误码	描述
0	成功
1005	请求参数错误
1012	玩家不存在

\*amount 不能为 0, **最小值为 0.01**

\*只保存成功交易（失败的交易不入库）

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "transactionId": "transaction_id",
  "playerId": "abcd123_usd",
  "amount": 230,
}
```

### 返回示例

```
{
  "data": {
    "transactionId": "transaction_id",
    "currency": 1006,
    "amount": 230,
    "balance": 1230,
  },
  "error": 0,
  "description": "Success",
}
```

## 4.3 转出额度

注意：

- ✓ **transactionId** 是每笔交易的唯一识别码。运营商需要为每笔转账请求发送新的 transactionId。
- ✓ 幂等\_对于重复请求，我方不对玩家钱包余额进行任何操作，但返回转账成功的响应。

### { APIDomain }/gi/v2/transferOut

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
transactionId	string	是	交易的 <b>唯一</b> 标识符
playerId	string	是	玩家 ID
amount	Decimal(10,2)	是	转出额度
transferAll	int	是	转出全部额度 (0 否, 1 是)
<b>Response – Object</b>			
transactionId	string	交易的 <b>唯一</b> 标识符	
currency	string	货币代码	
amount	Decimal(10,2)	本次转出额度（最大位数限制）	
balance	Decimal(10,2)	转账完成后的玩家余额（无论成功或失败）	

错误码	描述
0	成功
1005	请求参数错误
1014	交易不存在
1011	玩家余额不足

\*amount 不能为 0，**最小值为 0.01**；transferAll 为 true 时，无视 amount

\*玩家余额为 0 时，不执行转出操作，返回错误码“1011”

\*只保存成功交易（失败的交易不入库）

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "transactionId": "transaction_id",
  "playerId": "abcd123_usd",
  "amount": 1230,
  "transferAll": false,
}
```

### 返回示例

```
{
  "data": {
```

```

    "transactionId": "transaction_id",
    "currency": 1006,
    "amount": 230,
    "balance": 0,
  },
  "error": 0,
  "description": "Success",
}

```

## 4.4 查询玩家余额

**{ APIDomain }/gi/v2/getBalance**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
playerIds	String	是	玩家 ID 可以设置多个玩家 ID，每个玩家 ID 之间使用","分割，注意","为半角符号 最多可以查询 100 个
<b>Response – Object Array</b>			
playerId	string	玩家 ID	
currency	string	货币代码	
balance	Decimal(10,2)	玩家余额	

错误码	描述
0	成功

\*批量查询时，失败的用户信息为空，不返回“玩家不存在”

### 请求示例

示例：

```

{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "playerIds": "abcd123_usd,abcd456_usd",
}

```

### 返回示例

```

{
  "data": [
    {
      "playerId": "abc123_usd",
      "currency": "USD",
      "balance": 0,
    },
    {
      "playerId": "abc456_usd",
      "currency": "USD",
      "balance": 1210,
    }
  ],
  "error": 0,
  "description": "Success",
}

```

## 4.5 查询单笔转账

**{ APIDomain }/gi/v2/getTransaction**

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
transactionId	string	是	查询的交易
<b>Response – Object</b>			
transactionId	string	交易 ID	
playerId	string	玩家 ID	
transferType	integer	转入(1)/转出(2)	
currency	string	货币代码	
amount	Decimal(10,2)	转账额度	
balance	Decimal(10,2)	转账完成后余额	

错误码	描述
0	成功
1005	请求参数错误
1014	交易不存在

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "transactionId": "transaction_id",
}
```

#### 返回示例

```
{
  "data": {
    "transactionId": "transaction_id",
    "transferType": 1,
    "currency": "USD",
    "amount": 910,
    "balance": 1000,
  },
  "error": 0,
  "description": "Success",
}
```

## 5.通用接口

### 5.1 获取游戏列表

**{ APIDomain }/gi/v2/getGameList**

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
<b>Response – Object Array</b>			
gameId	int	游戏 ID (唯一)	
name	Object	游戏名称  格式为: { "en": "string", // 英文名 "zh": "string", // 中文名 }	
gameType	int	游戏类型(1.slot)	

provider	string	厂商: PG / JL
status	int	0 活跃 1 维护 2 不可用

错误码	描述
0	成功

### 请求示例

示例:

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
}
```

### 返回示例

```
{
  "data": [
    {
      "gameId": "31009",
      "name": {
        "en": "Fortune Tiger",
        "zh": "虎虎生财",
      },
      "gameType": 1,
      "provider": "PG",
      "status": 0
    },
    {
      "gameId": "31201",
      "name": {
        "en": "Fortune Tree",
        "zh": "发财树",
      },
      "gameType": 1,
      "provider": "JL",
      "status": 0
    },
  ],
  "error": 0,
  "description": "Success",
}
```



## 5.2 获取游戏记录

{ APIDomain }/gi/v2/getGameHistory

参数		类型	必需	说明
traceId		string	是	请求的唯一标识符
hash		string	是	哈希值
operatorId		string	是	商户 ID
provider		string	否	PG/JL, 按品牌筛选; 不填则不过滤
rowVersion		Long	是	数据版本号, 从 0 开始
Response – Object Array				
rowVersion		Long	数据版本号	
result	roundId	string	注单 ID (唯一)	
	parentRoundId	string	母注单 ID	
	playerId	string	玩家 ID	
	currency	string	币种	
	gameId	integer	游戏 ID	
	provider	string	厂商	
	betAmount	Decimal	投注额度	
	winAmount	Decimal	派彩额度	
	rtp	integer	本回合的 rtp 档位	
	isBuy	int	是否购买模式(0 否, 1 是)	
	isFeature	int	是否特殊旋转(0 否, 1 是)	
	isEnd	int	是否最后回合(0 否, 1 是)	
	isExtra	int	是否加注模式(0 否, 1 是)	
	settleTime	integer	结算时间	

错误码	描述
0	成功

\*如果返回的记录数少于 20,000 条, 请停止并等待一个时间间隔 (建议 5 分钟) 再进行下一次 API 请求

\*可通过检查每个调用中重复的 roundId 来识别重复的记录

\*每批记录数, 最大 20,000 条; 时间为秒级时间戳

\*支持 gzip1.1 压缩, HTTP 请求头: Accept-Encoding:gzip

### 请求示例

示例:

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "rowVersion": 0,
}
```

返回示例

```
{
  "data": {
    "rowVersion": 5000,
    "result": [
      {
        "roundId": "17272558760000000001",
        "parentRoundId": "17272558790000001259",
        "playerId": "abc123_usd",
        "currency": "USD",
        "gameId": 31219,
        "provider": "JL",
        "betAmount": 10,
        "winAmount": 0,
        "rtp": 0,
        "isBuy": false,
        "isFeature": false,
        "isEnd": true,
        "isExtra": false,
        "settleTime": 0
      }
    ],
    "error": 0,
    "description": "Success",
  }
}
```

5.3 设置玩家返奖率

{ APIDomain }/gi/v2/player/setRTP

参数	类型	必需	说明
traceld	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
playerIds	string	是	玩家 ID 可以设置多个玩家 ID，每个玩家 ID 之间使用","分割，注意","为半角符号 最多可以查询 100 个
rtp	int	是	0/1/2/3/4/5/6/99

			"99" 代表清除玩家身上的 rtp 设定
<b>Response – Object Array</b>			
playerId	string	玩家 Id	
rtp	int	玩家 RTP	

错误码	描述		
0	成功		

\*批量设置时，失败的用户信息为空，不返回“玩家不存在”

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "playerIds": "abcd123_usd,abcd456_usd",
  "rtp": 1,
}
```

### 返回示例

```
{
  "data": [
    {
      "playerId": "abc123_usd",
      "rtp": 1,
    },
    {
      "playerId": "abc456_usd",
      "rtp": 1,
    }
  ],
  "error": 0,
  "description": "Success",
}
```

## 5.4 查询玩家返奖率

**{ APIDomain }/gi/v2/player/getRTP**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符

hash	string	是	哈希值
operatorId	string	是	商户 ID
playerIds	string	是	玩家 ID  可以设置多个玩家 ID，每个玩家 ID 之间使用","分割，注意","为半角符号 最多可以查询 100 个
<b>Response – Object Array</b>			
playerId	string	玩家 ID	
rtp	int	0/1/2/3/4/5/6/99	

错误码	描述
0	成功

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "playerIds": "abcd123_usd,abcd456_usd",
}
```

### 返回示例

```
{
  "data": [
    {
      "playerId": "abc123_usd",
      "rtp": 1,
    },
    {
      "playerId": "abc456_usd",
      "rtp": 1,
    }
  ],
  "error": 0,
  "description": "Success",
}
```

## 5.5 创建玩家

**注意：**按 2.1->2.2 的流程，玩家帐户将被自动创建，所以此 API 的执行是非强制性的；也可以先创建玩家，然后调用 4.1 进入游戏（仅限转账钱包）

**{ APIDomain }/gi/v2/player/create**

参数	类型	必需	说明
traceId	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
playerId	string	是	玩家 ID ( <b>3-32 位, 字母,数字,@,-,_</b> )
currency	string	是	货币代码
<b>Response – Object</b>			
error	int	0	
description	string	success	

错误码	描述
0	成功
1005	请求参数错误

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "playerId": "abc123_usd",
  "currency": "USD",
}
```

### 返回示例

```
{
  "error": 0,
  "description": "Success",
}
```

## 5.6 踢出玩家

**{ APIDomain }/gi/v2/player/kick**

参数	类型	必需	说明
----	----	----	----

traceId	string	是	请求的唯一标识符
hash	string	是	哈希值
operatorId	string	是	商户 ID
playerId	string	是	玩家 ID
<b>Response – Object</b>			
error	int	0	
description	string	success	

错误码	描述
0	成功

\*将玩家从游戏中踢出，原 token 将失效；玩家可重新登录游戏

### 请求示例

示例：

```
{
  "traceId": "b3f37e57-2873-40b1aa95-f126c25ed311",
  "hash": "e10adc3949ba59abbe56e057f20f883e",
  "operatorId": "abc123",
  "playerId": "abc123_usd",
}
```

### 返回示例

```
{
  "error": 0,
  "description": "Success",
}
```

## 6.附录

### 6.1 错误码

错误码	描述	description
0	成功	Success
1001	内部服务器错误	Internal server error
1002	无效的运营商	Invalid operator
1003	运营商权限不足	Limited permissions
1004	无效的哈希码	Invalid hash code
1005	请求参数错误	Invalid parameters

1006	无效的玩家令牌	Invalid player token
1007	玩家令牌已过期	Player token expired
1008	游戏维护中	Game is under maintenance
1009	游戏不可用	Game is unavailable
1010	游戏不存在	Game not found
1011	玩家余额不足	Insufficient balance
1012	玩家不存在	Player not found
1013	投注异常（超时或参数异常）	Bet is pending
1014	交易不存在	Transaction not found
1015	货币代码错误或不支持	Invalid currency code
1016	无效请求	Invalid request
1017	玩家已存在	Player already existed
1018	交易已存在	Transaction Already Exists
1019	无效 IP	Invalid IP
1020	已达最大请求限制	Max Request Limit
1021	无效 RTP	Invalid RTP

## 6.2 货币

代码	货币名称	基础单位
AED	阿联酋迪拉姆	1
ARS	阿根廷比索	1
AUD	澳大利亚元	1
BDT	孟加拉塔卡	1
BHD	巴林第纳尔	1
BND	文莱元	1
BOB	玻利维亚诺	1
BRL	巴西雷亚尔	1
CAD	加拿大元	1
CHF	瑞士法郎	1
CLP	智利比索	1
CNY	人民币	1
COP	哥伦比亚比索	1000
CUP	古巴比索	1
EGP	埃及镑	1
ETB	埃塞俄比亚比尔	1
EUR	欧元	1
GBP	英镑	1
HKD	港元	1
IDR	印尼盾	1000
ILS	以色列新谢克尔	1

INR	印度卢比	1
JPY	日元	1
KES	肯尼亚先令	1
KHR	柬埔寨瑞尔	1000
KRW	韩元	1000
KWD	科威特第纳尔	1
LAK	老挝基普	1000
LKR	斯里兰卡卢比	1
MMK	缅甸元	1000
MNT	蒙古图格里克	1000
MOP	澳门元	1
MVR	马尔代夫拉菲亚	1
MXN	墨西哥比索	1
MYR	马来西亚令吉	1
MZN	莫桑比克梅蒂卡尔	1
NGN	尼日利亚奈拉	1
NPR	尼泊尔卢比	1
NZD	新西兰元	1
PEN	秘鲁索尔	1
PHP	菲律宾比索	1
PKR	巴基斯坦卢比	1
PYG	巴拉圭瓜拉尼	1000
QAR	卡塔尔里亚尔	1
RUB	俄罗斯卢布	1
SAR	沙特里亚尔	1
SEK	瑞典克朗	1
SGD	新加坡元	1
THB	泰铢	1
TRY	土耳其里拉	1
TWD	新台币	1
TZS	坦桑尼亚先令	1000
USD	美元	1
UYU	乌拉圭比索	1
VES	委内瑞拉玻利瓦尔	1
VND	越南盾	1000
XAF	中非法郎	1
XCD	东加勒比元	1
XOF	西非法郎	1
XPF	太平洋法郎	1
ZAR	南非兰特	1
TRX	波场币（虚拟货币）	1
TUSD	TrueUSD（虚拟货币）	1
USDC	USD Coin（虚拟货币）	1



USDT	泰达币（虚拟货币）	1
------	-----------	---

## 6.3 语言

代码	语言	PG	JILI
en	英文（默认）	√	√
da	丹麦文	√	√
de	德文	√	√
es	西班牙文	√	√
fi	芬兰文	√	
fr	法文	√	√
id	印尼文	√	√
it	意大利文	√	√
ja	日文	√	√
ko	韩文	√	√
nl	荷兰文	√	√
no	挪威文	√	
pl	波兰文	√	
pt	葡萄牙文	√	√
ro	罗马尼亚文	√	√
ru	俄文	√	√
sv	瑞典文	√	√
th	泰文	√	√
tr	土耳其文	√	√
vi	越南文	√	√
zh	中文	√	√
my	缅甸文	√	√
hi	印地文	√	√
ur	乌尔都文	√	√
bn	孟加拉文	√	√
el	希腊文	√	√
ar	阿拉伯文	√	
lo	老挝文	√	
uk	乌克兰文	√	
ta	泰米尔文		√
ms	马来文		√

\*如语言不支持，游戏将默认显示为英文(en)

## 6.4 返奖率

代码	返奖率(RTP)	说明
0	95~97% (默认)	中~高波动
1	1~20%	低波动
2	20~40%	低波动
3	40~60%	低波动
4	60~80%	低波动
5	85~89%	中波动
6	90~94%	中波动
99	代表玩家身上无返奖率设定	新玩家默认值

### 返奖率设置：

- ✓ 运营商可在后台调整游戏的默认返奖率，也可通过 api 接口给玩家设定返奖率
- ✓ 新用户身上无返奖率设定，按游戏默认返奖率执行
- ✓ 设置玩家返奖率后，按玩家返奖率执行
- ✓ 清除掉玩家身上的返奖率后（代码"99"），按游戏默认返奖率执行

**注：**返奖率是指玩家在游戏过程中的预期回报，需要一定的游戏局数才能收敛至期望值。

## 6.5 须知

### 货币单位：

运营商在提交对接申请时，如有加入下述货币单位为 1000 的币种，须选择是由运营商或是由游戏商对下述货币进行单位转换：

- COP（哥伦比亚比索）
- IDR（印度尼西亚卢比盾）
- KHR（柬埔寨瑞尔）
- KRW（韩元）
- LAK（老挝基普）
- MMK（缅甸元）
- MNT（蒙古图格里克）
- PYG（巴拉圭瓜拉尼）
- TZS（坦桑尼亚先令）
- VND（越南盾）

### 单位转换关系：

1. 运营商转换：运营商以 k 为单位传入，游戏商以 k 为单位返回（最小额度 0.01）
2. 游戏商转换：运营商以 1 为单位传入，游戏商以 1 为单位返回（最小额度 10）
3. 游戏内显示、报表、结算，均以 k 为单位