

# The Gaming Platform

## TGP 运营商 API 集成指南



**TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB**

Copyright © 2016 TGP Asia Limited. All Rights Reserved. This document contains  
confidential and proprietary information of TGP Asia Limited.

# 目录

1. [介绍](#)
2. [背景](#)
  - 2.1 [使用者群](#)
  - 2.2 [定义与条款](#)
3. [概述](#)
4. [服务与格式](#)
5. [安全和授权](#)
  - 5.1 [API调用的OAuth授权](#)
6. [玩家身份验证](#)
  - 6.1 [顺序图](#)
  - 6.2 [游戏大厅端点](#)
  - 6.3 [玩家身份验证](#)
  - 6.4 [玩家授权取消](#)
7. [外部电子钱包集成](#)
  - 7.1 [顺序图](#)
  - 7.2 [重试\(Retries\)](#)
  - 7.3 [取得余额\(GetBalance\)](#)
  - 7.4 [电子钱包加钱\(WalletCredit\)](#)
  - 7.5 [电子钱包扣钱\(WalletDebit\)](#)
  - 7.6 [报告方法\(Reporting Methods\)](#)
    - 7.6.1 [转账历史\(Transfer History\)](#)
    - 7.6.2 [投注交易历史\(Bet Transaction History\)](#)
    - 7.6.3 [游戏历史\(Game History\)](#)
    - 7.6.4 [投注历史\(Bet History\)](#)
    - 7.6.5 [来自游戏供应商的游戏历史](#)
8. [单一电子钱包API集成](#)
  - 8.1 [顺序图](#)
  - 8.2 [客户端服务器授权](#)
  - 8.3 [扣钱\(Debit\)](#)
  - 8.4 [加钱\(Credit\)](#)
  - 8.5 [取消\(Cancel\)](#)
  - 8.6 [取得余额\(GetBalance\)](#)
  - 8.7 [错误和重试处理\(Errors and Retry\)](#)

## 9. [游戏启动器](#)

### [9.1 游戏列表API](#)

### [9.2 游戏启动页](#)

### [9.3 启动游戏的示例](#)

## [附录](#)

### [附录A - OAuth错误表](#)

### [附录B - 支持HTTP的状态代码](#)

### [附录C - VIP级别](#)

### [附录D - 语言](#)

### [附录E - 货币](#)

### [附录F - 交易类型](#)

### [附录G - 奖金类型](#)

### [附录H - 游戏类型](#)

### [附录I - 外部电子钱包集成的错误表](#)

### [附录J - 单一电子钱包集成的错误表](#)

### [附录K - 游戏供应商列表](#)

### [附录L - 平台类型](#)

### [附录M - 平衡同步API](#)

## [修订记录](#)

# 1. 介绍

本文档针对运营商与游戏平台(TGP)的集成提供所需的技术性作业概述。

本文档详细描述信息交换、信息结构、方法调用及参数在API界面所使用的方法列表、response的返回代码、和request及response的应用范例。

# 2. 背景

## 2.1 使用者群

本文档目标是技术用户群，并假设这用户具备相关的基本知识，包括互联网服务、互联网通信协议、和编程语言的能力。

## 2.2 定义与条款

术语	定义
UGS	统一游戏服务器(Unified Gaming Server)，可与"TGP"互换使用。
Operator	游戏网站运营商，针对TGP授权。亦称权证持有者(Licensee)。
Game provider	第三方供应商，提供与TGP集成的游戏。
BO	后台(Backoffice)
Player	游戏运营商的客户，亦是TGP游戏大厅(lobby)和游戏的使用者。

## 3. 概述

UGS是一个远程游戏服务器(RGS)，为多个游戏供应商提供单一的集成点。运营商需执行两个主要的作业流程来和TGP集成。

### 1. 玩家身份验证流程

- 从运营商网站玩TGP游戏，运营商必须将玩家资讯并伴随一个身份验证令牌(token)送到TGP游戏大厅。此令牌亦由服务器端送到TGP，以便在玩家进入游戏大厅时予以身份辨识。
- TGP使用运营商送来的令牌辨识进入的玩家。
- 一旦通过身份验证，玩家可以玩任何TGP游戏。

### 2. 财务流程

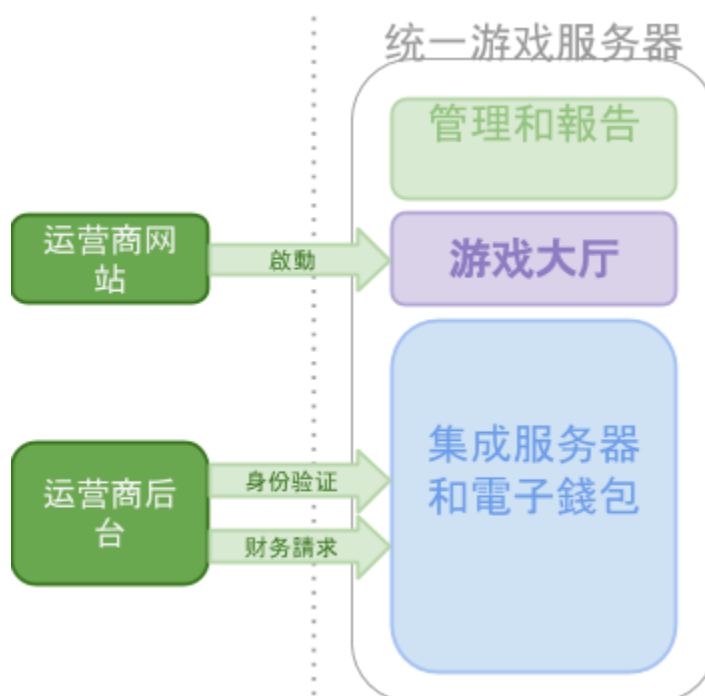
#### a. 外部电子钱包(External Wallet)

- TGP针对每一个玩家提供一个电子钱包。玩家必须在这电子钱包存钱后，才可在平台上玩游戏。
- 转进(fund-in)、转出(fund-out)、和余额请求(balance requests)是由运营商在UGS上启动。

#### b. 单一电子钱包(Single Wallet)

- TGP不会帮玩家处理电子钱包，但会将相关操作请求转发给运营商那端的API。
- 电子钱包请求是由UGS对运营商发出。

UGS/TGP集成概述图



## 4. 服务与格式

TGP采用REST-ful API，但并不严格遵守所有的REST原则。欲知更多REST的信息，请参阅RFC 2616。

所有的服务调用(service calls)都使用POST或GET动词以及JSON格式启动请求和响应(有些响应是CSV格式)。GET请求会将数据在查询字符串中送出，POST请求数据则在请求主体(request body)中以JSON格式送出。

设定HTTP Header Accept (例如“application/json”)可显示想要的response内容型态。送出request时，客户端(client)需要设置Content-Type HTTP Header (eg: “Content-Type: application/json”)。

**request范例：**

GET /api/player/balance?userid=fa39953d&cur=RMB

**response范例：**

```
{
  "cur": "RMB",
  "bal": 550
}
```

### 4.1 错误处理

一般而言，当API方法的响应之HTTP状态代码不是200时，就代表发生了错误，且API客户端可在返回的响应中检视错误属性，以取得错误代码和信息，如下表所示。

参数	定义	必须有?	类型
err	大于零的错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

## 5. 安全和授权

为了确保运营商后台系统和TGP之间的安全得以强制执行，使用OAuth2通信协议来验证运营商。

为了确保玩家连结的安全，一个由运营商产生的令牌会被用来做玩家身份验证。

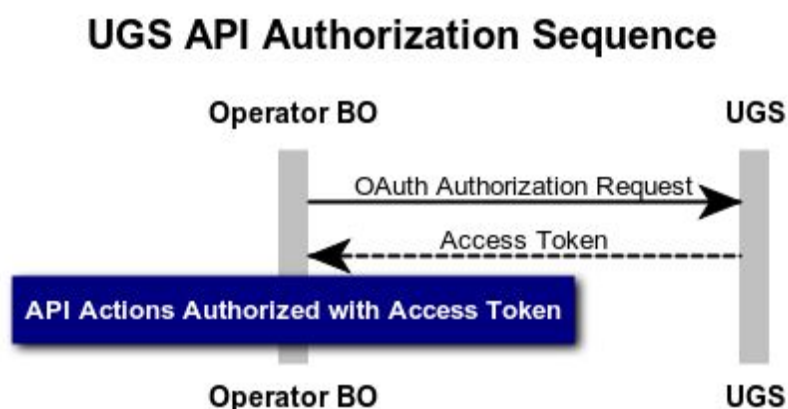
所有与TGP服务器对服务器的通信会对IP地址加以限制，且必须遵守SSL安全协议。

### 5.1 API调用的OAuth授权

TGP API会要求OAuth授权，用来验证运营商已获得通信授权。

为了实现OAuth 2.0标准，会依据一个共享的客户端标识码(Client ID)和客户端密钥(Client Secret)产生一个访问令牌(Access Token)，并在通信创建时提供给运营商。然后，所有的服务器API请求皆会将此访问令牌当作Authorization HTTP Header的Bearer值。

访问令牌的默认有效时间为1小时，一旦过其，必须要刷新才可继续执行API调用。



## HTTP方法：

POST

## HTTP header:

Content-Type: application/x-www-form-urlencoded

## URL方法：

http://<TGPIntegrationAPIServer>/api/oauth/token

## 方法参数：

参数	定义	必须有?	类型
client_id	运营商的专有标识码，在通信创建时提供。	是	类型：字符串 最大长度：50字符。
client_secret	运营商的密钥，在通信创建时提供。	是	类型：字符串 最大长度：200
grant_type	永远是“client_credentials”。	是	类型：字符串
scope	永远是“playerapi”。	是	类型：字符串

## 返回参数：

参数	定义	必须有?	类型
access_token	调用运营商API服务器的访问令牌。	是	类型：字符串 最大长度：255字符。
token_type	永远是“bearer”。	是	类型：字符串
expires_in	时间终止以秒计算，共3600秒(1小时)。	是	类型：整数
scope	永远是“playerapi”。	是	类型：字符串



### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：字符串
err_description	以英文描述的错误说明。	否	类型：字符串

### Calls范例：

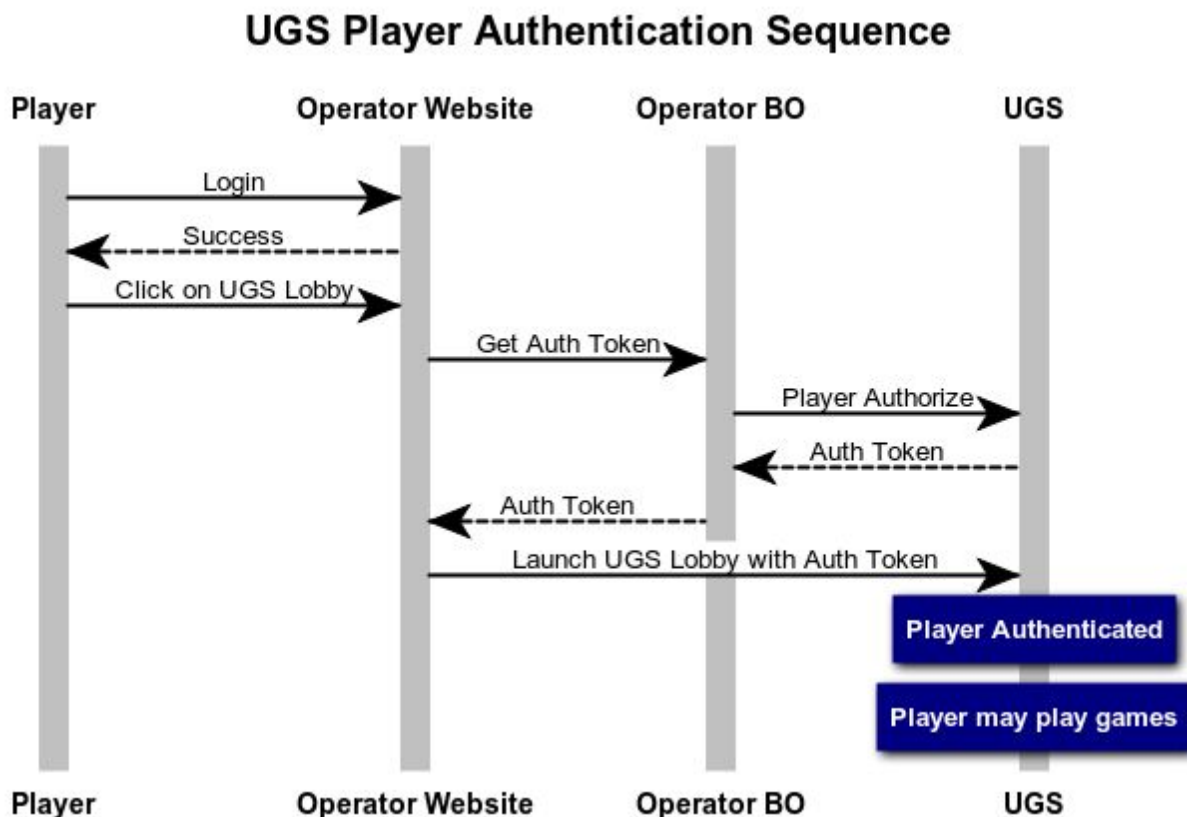
请求(Request)
<p>POST /api/oauth/token HTTP/1.1</p> <p>HEADER:</p> <p>Content-Type: application/x-www-form-urlencoded</p> <p>CONTENT:</p> <p>grant_type=client_credentials&amp;scope=playerapi&amp;client_id=Operator-Integration&amp;client_secret=1uuvtpXapvS6RwslyA2FgrZQA7V5EoPNT8djr0o8wqf1</p>
响应(Response)
<p><b>请求成功的响应结果：</b></p> <pre>{   "access_token": "jk123j42kgs5dghdg...",   "token_type": "bearer",   "expires_in": 3600,   "scope": "playerapi" }</pre> <p><b>请求失败的响应结果：</b></p> <pre>{   "error": "invalid_scope",   "error_description": "Invalid Scope" }</pre> <p>欲知OAuth错误代码，请参阅<a href="#">附录A</a>。</p>

## 6. 玩家身份验证

为了验证玩家可进入TGP游戏大厅，运营商会要求一个授权令牌，并在启动游戏大厅时将它放在查询字符串中传递。

### 6.1 顺序图

身份验证程序在下图中描述：



title UGS玩家身份验证程序

玩家->运营商网站：登录

运营商网站->玩家：成功

玩家->运营商网站：单击UGS游戏大厅

运营商网站->运营商后台系统获取身份验证令牌

运营商后台系统->UGS：玩家授权

UGS->运营商后台系统：身份验证令牌

运营商后台系统->运营商网站身份验证令牌

运营商网站->UGS：以身份验证令牌启动UGS大厅

Note over UGS：通过身份验证的玩家

Note over UGS：玩家可以开始玩游戏

## 6.2 游戏大厅端点

### 端点网址示例：

一个大厅端点会在通信创建时提供给运营商。

```
http://dev.tgp-server-example.com/{lobby_path}?token={authtoken}
```

要启动TGP大厅，运营商会产生大厅路径取代{lobby\_path}以及认证令牌取代{authtoken}，并由TGP服务器验证其有效性。

注：可在TGP后台网站检视{lobby\_path}。

## 6.3 玩家身份验证

运营商调用玩家身份认证方法以获得一个身份验证令牌，并将更新的玩家信息传给TGP。若此指定的玩家帐户在TGP系统中不存在，就会创建一个新的账户。

当此方法被调用时，系统假设玩家在运营商系统中是活跃的，因为TGP不会检查它在运营商的状态。若运营商要关闭一个玩家连结，必须调用玩家取消授权(Player Deauthorize)方法。

### 重要：

- 调用此API时，请注意务必传递正确的'istestplayer'参数值。若针对玩家'Michael'传递'istestplayer=False'，那么玩家'Michael'就会在TGP系统中以真实玩家身份被创建，且无法经由API将此身份变更为测试玩家。欲知更详细说明，请参阅下面的参数表，且若有任何疑问，可询问TGP团队。

### HTTP方法：

POST

### URL方法：

```
http://<TGPIntegrationAPIServer>/api/player/authorize
```

### 方法参数：

参数	定义	必须有?	类型
ipaddress	玩家客户端IP地址。TGP会验证客户端的IP地址是否被变更。若被变更，玩家登入时尝试使用authtoken就会失败。	是	类型：字符串 最大长度：15字符。
username	用来显示玩家的友善名称。此名称可重复。 ex: james_johnson	是	类型：字符串 最大长度：50字符。
userid	玩家唯一的辨识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度：50字符。

### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

tag	<p>有效的JSON格式文字，可用来包含关于玩家的诠释数据(metadata)。此字段可包含于玩家报告中。</p> <p>示例：</p> <pre>"{"aff":"123","tag1":"abc"}"</pre> <p>注：此参数传递的键值会以字典型态储存。若不确定要传递什么，将其留空。若它是无效的JSON，就会产生错误。</p>	否	<p>类型：字符串(JSON)</p> <p>最大长度：300</p>
lang	<p>玩家所使用的语言代码。</p>	是	<p>类型：字符串(ISO-639-1和ISO3166-1)，请参阅附录D</p> <p>长度：5字符。</p>
cur	<p>玩家玩游戏所使用的货币(ISO 3 char code)，</p>	是	<p>类型：字符串(ISO 4217 3字符代码)</p> <p>长度：3字符。</p>
betlimitid	<p>玩家的VIP等级，用来设置投注限制。</p> <p>请参阅附录C的查找表。</p>	是	<p>类型：整数</p>
istestplayer	<p>玩家在运营商这一端是否为测试玩家-亦称为庄家玩家(House Player)。</p> <p><b>重要：</b> 针对测试玩家，请务必设定此参数为True，因为一旦玩家在TGP数据库中创建后，即不能更改此值，也就是说，测试玩家无法经由API转换为真实玩家，反之亦然。</p> <p>若未提供此参数，其默认值为False，但不建议这样做。底线是：永远都要给这参数一个适当的值。</p>	是	<p>类型：逻辑(true或false)</p>
platformtype	<p>玩家的平台。</p> <p>默认值为0 (桌面计算机)。</p> <p>请参阅<a href="#">附录L-平台类型</a>的所有值。</p>	否	<p>类型：整数</p>
loginurl	<p>当TGP身份验证失败时，返回玩家的网址。若未提供此参数，将会使用TGP组态中的设定值。</p>	否	<p>类型：字符串</p> <p>最大长度：255字符。</p>
cashierurl	<p>将玩家送去补充余额的网址。若未提供此参数，将会使用TGP组态中的设定值。</p>	否	<p>类型：字符串</p> <p>最大长度：255字符。</p>
helpurl	<p>将玩家送到客服的网址。若未提供此参数，将会使用TGP组态中的设定值。</p>	否	<p>类型：字符串</p> <p>最大长度：255字符。</p>
termsurl	<p>将玩家送到服务条款的网址。若未提供此参数，将会使用TGP组态中的设定值。</p>	否	<p>类型：字符串</p> <p>最大长度：255字符。</p>

#### 返回：

参数	定义	必须有?	类型
<b>authtoken</b>	运营商所产生的授权令牌，是对玩家连结的唯一辨识。	是	类型：字符串 最大长度：2000字符。
<b>isnew</b>	是否为TGP尚未遇到过的新玩家。	是	类型：逻辑(true或false)

#### 错误参数：

参数	定义	必须有?	类型
<b>err</b>	大于零的错误代码。	否	类型：整数
<b>errdesc</b>	以英文描述的错误说明。	否	类型：字符串

## Calls范例 :

### 请求(Request)

#### POST网址 :

http://<TGPIntegrationAPIServer>/api/player/authorize

#### HEADER:

Accept: application/json

Authorization: Bearer UG0zRtJH9Zd7oqQCMVs5ux0ky1zyuT0vtJWLK...(trimed)

#### BODY:

```
{
  "ipaddress": "123.123.1.95",
  "username": "john",
  "userid": "12345",
  "lang": "zh-CN",
  "cur": "RMB",
  "betlimitid": 5,
  "platformtype": 1,
  "istestplayer": false
}
```

### 响应(Response)

#### 请求成功的响应结果 :

```
{
  "authtoken": "j9r1VRbwRsKG0pE2l55fn8Cw5TF2guj81tSr...(trimmed)",
  "isnew": false
}
```

## 6.4 玩家授权取消

运营商可调用玩家授权取消方法，移除特定玩家所有活动中的连结。

例如，当玩家从运营商的会员网站登出时，运营商的网页程序就应该要调用此方法以通知TGP，如此玩家就无法在浏览器的浏览记录中挑选一个网址进入TGP游戏大厅。

TGP会在创建通信时，提供运营商此方法网址。

### HTTP方法：

POST

### URL方法：

`http://<TGPIntegrationAPIServer>/api/player/deauthorize`

### 方法参数：

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的辨识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度： 50字符。

### 返回：

参数	定义	必须有?	类型
<b>success</b>	是否已更新任何令牌。	是	类型：逻辑：(true或false)
<b>desc</b>	以英文描述的授权取消结果。	是	类型：字符串 最大长度： 2000字符。

### 错误参数：

参数	定义	必须有?	类型
<b>err</b>	大于零的错误代码。	否	类型：整数
<b>errdesc</b>	以英文描述的错误说明。	否	类型：字符串

## Calls范例 :

### 请求(Request)

#### POST网址 :

http://<TGPIntegrationAPIServer>/api/player/deauthorize

#### HEADER:

Authorization: Bearer UG0zRtJH9Zd7oqQCMVs5ux0ky1zyuT0vtJWLK...(trimmed)

#### CONTENT:

```
{
  "userid": "0586cecf-4787- 49c5-8c35-2118bfa10913"
}
```

### 响应(Response)

#### 请求成功的响应结果 :

```
{
  "Success": true,
  "desc": "Tokens deauthorized."
}
```



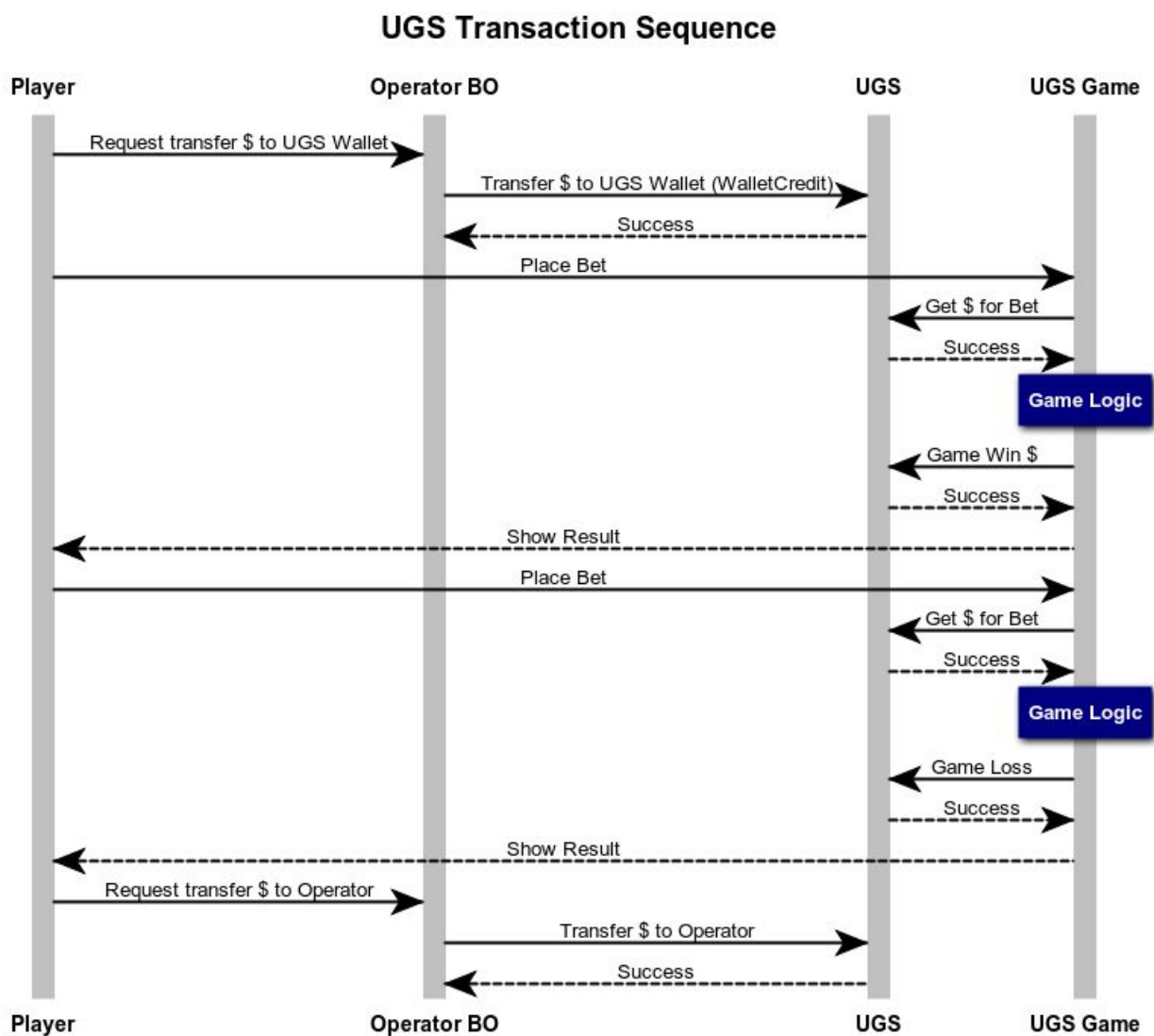
## 7. 外部电子钱包集成

财务方法支持TGP系统中资金从玩家电子钱包转入和转出，并对运营商提供玩家的财务信息。

使用外部电子钱包集成方法系假设运营商会为TGP电子钱包系统执行转进/转出的功能。这不同于单一电子钱包，使用运营商的电子钱包，请参阅第8节的相关说明。

### 7.1 顺序图

下图描述了财务方法的调用序列：



title UGS交易顺序

玩家->运营商后台系统：请求将钱转到UGS电子钱包

运营商后台系统-> UGS：将钱转到UGS电子钱包(WalletCredit )

UGS ->运营商后台系统：成功

玩家-> UGS游戏：下注

UGS游戏-> UGS：取得投注的钱

UGS -> UGS游戏：成功

note over UGS游戏：游戏逻辑

UGS游戏-> UGS：游戏赢钱

UGS -> UGS游戏：成功

UGS游戏->玩家：显示结果

玩家-> UGS游戏：下注

UGS游戏-> UGS：取得投注的钱

UGS -> UGS游戏：成功

note over UGS游戏：游戏逻辑

UGS游戏-> UGS：游戏输钱

UGS -> UGS游戏：成功

UGS游戏->玩家：显示结果

玩家->运营商后台系统：请求将钱转到运营商

运营商后台系统-> UGS：将钱转到运营商(WalletDebit)

UGS ->运营商后台系统：成功

## 7.2 重试 (Retries)

### 重试

若调用因为网络或其它问题而失败，此交易会不断的重试。若收到重复的交易标识码，TGP API会回复相同的结果，除了余额字段外，因为这字段会在每一次调用时作实时的更新显示。一个重复旗标“dup”会添加到重复调用的响应。此重复旗标不会添加到GET相关的方法，如GetBalance方法。

在这种情况下，运营商应以下列模式重试此调用：

尝试次数	自最后一次尝试的时间
1	立即
2	立即
3	5秒
4	15秒
5	45秒
6	135秒
7	405秒
8	1215秒
9	3645秒

在经过第9次的尝试后，运营商应每隔3645秒重试一次。

## 7.3 取得余额 (GetBalance)

**HTTP方法：**

GET

**URL方法：**

http://<TGPIntegrationAPIServer>/api/player/balance

**方法参数：**

参数	定义	必须有?	类型
userid	玩家唯一的辨识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度：50字符。
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。

**返回：**

参数	定义	必须有?	类型
bal	玩家的余额量。	是	类型：小数点 精确度：2位数。
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。

**错误参数：**

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

## Calls范例 :

请求(Request)	响应(Response)
<p><b>调用 :</b></p> <pre>http://&lt;TGPIntegrationAPIServer&gt;/player/balance?userid=fa39953d&amp;cur=RMB</pre>	<p><b>请求成功的响应结果 :</b></p> <pre>{     "bal": 550,     "cur ": "RMB" }</pre>

## HTTP方法：

GET

## URL方法：

http://<TGPIIntegrationAPIServer>/api/players/balance/total

请注意，这是多个玩家，而非一个玩家。

## 说明：

此方法返回以货币为群组的玩家总余额。

## 方法参数：

参数	定义	必须有?	类型
<b>includetestplayers</b>	是否包含测试玩家； 默认值为false。	否	类型：逻辑  eg. true

## 返回：

参数	定义	必须有?	类型
<b>cur</b>	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。
<b>amt</b>	该货币的总余额。	是	类型：小数点 精确度：2位数。

响应格式：csv。

## HTTP方法：

GET

## URL方法：

http://<TGPIIntegrationAPIServer>/api/players/balance/list

请注意，这是多个玩家，而非一个玩家。

## 说明：

获取所有玩家的余额。

## 方法参数：

参数	定义	必须有？	类型
<b>cur</b>	玩家用于TGP系统的货币(ISO 3字符代码)。 若未指定，则将返回所有货币。	否	类型：字符串(ISO 3字符代码) 长度：3字符。
<b>includezero</b>	是否包括余额为零； 默认值为false。	否	类型：逻辑 eg. True
<b>includetestplayers</b>	是否包含测试玩家； 默认值为false。	否	类型：逻辑 eg. True

## 返回：

参数	定义	必须有？	类型
<b>userid</b>	玩家唯一的标识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度：50字符。
<b>username</b>	玩家名称。	是	类型：字符串 最大长度：50字符。
<b>amt</b>	该货币的总余额。	是	类型：小数点 精确度：2位数。
<b>cur</b>	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。

返回格式：csv。

## TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

## 7.4 电子钱包加钱 (WalletCredit)

**HTTP方法：**

POST

**URL方法：**

http://<TGPIIntegrationAPIServer>/api/wallet/credit

**方法参数：**

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的辨识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度：50字符。
<b>amt</b>	玩家在TGP电子钱包增添的金额；必须是正数。	是	类型：小数点 精确度：小数点两位
<b>cur</b>	请求转账的货币(ISO 3字符代码)，若与玩家在TGP的货币不符，就会产生错误。	是	类型：字符串(ISO 3字符代码) 长度：3字符。
<b>txid</b>	特定动作的辨识码。  若遇到重复的txid，且若所有其它的参数都相同，前一次的结果会和重复旗标(Duplicate flag)一起被重送。 若这些参数不符，则会引发错误。	是	类型：字符串 最大长度：50字符。
<b>timestamp</b>	有时区的时间戳。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
<b>desc</b>	交易的延伸描述。	否	类型：字符串 最大长度：255字符。



### 返回：

参数	定义	必须有?	类型
bal	交易完成后的玩家余额。	是	类型：小数点 精确度：小数点两位
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。
txid	特定动作的标识码。反映传入的txid方法参数值。	是	类型：字符串 最大长度：50字符。
ptxid	由此动作产生的TGP平台交易标识码。	是	类型：全局唯一标识符 长度：36字符。
dup	重复旗标，显示是否为重送的结果。	否	类型：逻辑 长度：1(1或0)

### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

## Calls范例 :

### 请求(Request)

#### 调用 :

POST http://<TGPIntegrationAPIServer>/api/wallet/credit HTTP/1.1

#### HEADER:

Authorization: Bearer  
ZaE3R2mB2F5QpFRWRsQdth8iYhRhfwTG6TQoZZyTb1ma3AlPgrRgC6dSFRRq9yAUi  
Content-Type: application/json; charset=utf-8

#### CONTENT:

```
{
  "userid": "0802a4d4-503c-45c2-baca-1c68fa9844b1",
  "amt": 100.0,
  "cur": "RMB",
  "txid": "5bb70123-ae1f-4e3e-9c40-9f35a98d3e8d",
  "timestamp": "2015-07-14T23:33:09+00:00",
  "desc": null
}
```

### 响应(Response)

#### 请求成功的响应结果 :

HTTP/1.1 200 OK  
Pragma: no-cache  
Cache-Control: no-cache  
Date: Tue, 14 Jul 2015 23:33:09 GMT  
Server: Microsoft-IIS/8.0  
X-AspNet-Version: 4.0.30319  
X-Powered-By: ASP.NET  
Content-Length: 131  
Content-Type: application/json; charset=utf-8  
Expires: -1

```
{
  "bal": 201.0,
  "cur": "RMB",
  "txid": "5bb70123-ae1f-4e3e-9c40-9f35a98d3e8d",
  "ptxid": "02ac4de6-d813-4dff-ac45-26d54db67725",
  "dup": false
}
```

## 7.5 电子钱包扣钱 (WalletDebit)

**HTTP方法：**

POST

**URL方法：**

`http://<TGPIntegrationAPIServer>/api/wallet/debit`

**方法参数：**

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的标识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度： 50字符。
<b>amt</b>	从TGP电子钱包的扣款金额；必须是正数。	是	类型：小数点 精确度：小数点两位
<b>cur</b>	请求转账的货币(ISO 3字符代码)，若与玩家在TGP的货币不符，就会产生错误。	是	类型：字符串(ISO 3字符代码) 长度： 3字符。
<b>txid</b>	特定动作的标识码。  若遇到重复的txid，且若所有其它的参数都相同，前一次的结果会和重复旗标(Duplicate flag)一起被重送。 若这些参数不符，则会引发错误。	是	类型：字符串 最大长度： 50字符。
<b>timestamp</b>	有时区的时间戳。	是	类型：日期(ISO) 长度： 25字符。  eg. 2014-09-25T07:07:31+00:00
<b>desc</b>	延伸的描述或注记。	否	类型：字符串 最大长度： 255字符。

### 返回：

参数	定义	必须有?	类型
bal	交易完成后的玩家余额。	是	类型：小数点 精确度：小数点两位
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	是	类型：字符串(ISO 3字符代码) 长度：3字符。
txid	特定动作的标识码。反映传入的txid方法参数值。	是	类型：字符串 最大长度：50字符。
ptxid	由此动作产生的TGP平台交易标识码。	是	类型：全局唯一标识符 长度：36字符。
dup	重复旗标，显示是否为重送的结果。	否	类型：逻辑 长度：1(1或0)

### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

## Calls范例 :

### 请求(Request)

#### 调用 :

POST http://<TGPIntegrationAPIServer>/api/wallet/debit HTTP/1.1

#### HEADER:

Authorization: Bearer

ZaE3R2mB2F5QpFRWRsQdth8iYhRhfwTG6TQoZZyTb1ma3AlPgrRgC6dSFRRq9yAUi

Content-Type: application/json ; charset=utf-8

#### CONTENT:

```
{
  "userid": "0802a4d4-503c-45c2-baca-1c68fa9844b1",
  "amt": 100.0,
  "cur": "RMB",
  "txid": "6063c527-4440-4adb-9b1c- 3c98f829ae30",
  "timestamp": "2015-07-14T23:37:37+00:00",
  "desc": null
}
```

### 响应(Response)

#### 请求成功的响应结果 :

HTTP/1.1 200 OK

Pragma: no-cache

Cache-Control: no-cache

Date: Tue, 14 Jul 2015 23:37:37 GMT

Server: Microsoft-IIS/8.0

X-AspNet-Version: 4.0.30319

X-Powered-By: ASP.NET

Content-Length: 131

Content-Type: application/json; charset=utf-8

Expires: -1

```
{
  "bal": 101.0,
  "cur": "RMB",
  "txid": "6063c527-4440-4adb-9b1c-3c98f829ae30",
  "ptxid": "b326612b-b500-4afa-a9bf-346c70925f66",
  "dup": false
}
```

## 7.6 报告方法 (Reporting Methods)

所有的报告方法都会以csv格式返回数据。

有4个报告方法：

- history/transfers
- history/bettransaction
- history/game
- history/bets

若要取得玩家的投注历史，使用history/bets方法大概是最佳的选择。

### 7.6.1 转账历史 (Transfer History)

这方法用来取得玩家转进和转出的历史。

**HTTP方法：**

GET

**内容类型(必需)：**

text/csv

**URL方法：**

`http://<TGPIntegrationAPIServer>/api/history/transfers`

## 方法参数：

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的标识代码。 ex: fc533ace-bb15 若未指定userid，所有日期区间内的记录都会被返回。	否	类型：字符串 最大长度：50字符。
<b>startdate</b>	转账的开始日期。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:00:00%2B00:00
<b>enddate</b>	唯一的转账结束日期。 'enddate'不得超过'startdate' 48小时。 注意：这是唯一的，因此若要在2014-09-25 08:00这时间点取得一小时的数据，'startdate'和 'enddate'应该分别是 2014-09-25T08:00:00%2B00:00 和 2014-09-25T09:00:00%2B00:00。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T08:00:00%2B00:00
<b>includetestplayers</b>	是否包含测试玩家； 默认值为false。	否	类型：逻辑  eg. true

## 返回：

参数	定义	类型
txid	转账交易标识码。这与运营商在加钱和扣钱作业中所提供的txid完全相同。	类型：字符串 最大长度：50字符。
timestamp	在TGP服务器上的转账时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
userid	玩家唯一的辨识代码。	类型：字符串 最大长度：50字符。
username	玩家的显示名称。	类型：字符串 最大长度：50字符。
amt	转账金额。 正数代表转账到TGP服务器(转进)； 负数代表从TGP服务器转出(转出)。	类型：小数点 精确度：2位数。
postbal	转账后玩家的余额。	类型：小数点 精确度：2位数。
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	类型：字符串(ISO 3字符代码) 长度：3字符。
playertype	用来辨识是否为测试玩家或真实玩家。请参阅下面数值的意义：  1=真实玩家 2=运营商测试玩家	类型：整数

响应格式：csv。

## 错误参数：

参数	定义	必须有？	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

## TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB



## Calls范例 :

### 请求(Request)

#### 调用 :

```
http://<TGPIIntegrationAPIServer>api/history/transfers?userid=fa39953d&startdate=2015-08-14T07:07:31%2B00:00&enddate=2014-08-15T07:07:31%2B00:00
```

### 响应(Response)

#### 请求成功的响应结果 :

```
txid,timestamp,userid,username,amt,postbal,cur,playertype
6063c527,2015-08-14T08:07:31+00:00,fa39953d,dave,200.00,400.00,RMB,1
5163d332,2014-08-14T08:08:31+00:00,fa39953d,dave,100.00,300.00,RMB,1
```

## 7.6.2 投注交易历史 (Bet Transaction History)

此方法可用来取得玩家的投注交易历史。

**HTTP方法：**

GET

**内容类型(必需)：**

text/csv

**URL方法：**

`http://<TGPIntegrationAPIServer>/api/history/bettransaction`

**方法参数：**

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的识别代码。 ex: fc533ace-bb15 若未指定userid，所有日期区间内的记录都会被返回。	否	类型：字符串 最大长度：50字符。
<b>startdate</b>	交易记录的开始日期。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T00:00:00%2B00:00
<b>enddate</b>	唯一的交易记录的结束日期。 'enddate'不得超过'startdate' 48小时。 注意：这是唯一的，因此若要在2014-09-25 08:00这时间点取得一小时的数据，'startdate'和'enddate'应该分别是 2014-09-25T08:00:00%2B00:00和 2014-09-25T09:00:00%2B00:00。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T01:00:00%2B00:00
<b>includetestplayers</b>	是否包含测试玩家； 默认值为false。	否	类型：逻辑  eg. True

返回：

参数	定义	类型
txid	来自游戏供应商的交易或投注标识码。	类型：字符串 最大长度：50
ugstxid	UGS内部交易唯一的标识码。	全局唯一标识符 最大长度：36字符。
timestamp	在TGP服务器上的转账时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
userid	玩家唯一的标识码。	类型：字符串 最大长度：50字符。
username	玩家的显示名称。	类型：字符串 最大长度：50字符。
amt	交易金额。 正数金额代表TGP电子钱包的加钱。关联的操作类型包括：  WinBet AdjustBet CancelTransaction FreeBet  负数金额代表TGP电子钱包的扣钱。关联的操作类型包括： PlaceBet AdjustBet CancelTransaction LoseBet  金额为零代表损失，关联的操作类型是： LoseBet	类型：小数点 精确度：2位数。
postbal	转账后玩家的余额。	类型：小数点 精确度：2位数。
cur	玩家用于TGP系统的货币(ISO 3字符代码)。	类型：字符串(ISO 3字符代码) 长度：3字符。
roundid	交易执行回合(round)时的游戏供应商标识码。	类型：字符串 最大长度：50字符。

#### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

<b>roundstatus</b>	报告回合时的状态。可为下列其中之一： Open Closed	
<b>actiontype</b>	动作类别，可为下列其中之一： PlaceBet WinBet AdjustBet CancelTransaction LoseBet FreeBet	类型：字符串 最大长度：20字符。
<b>gameprovider</b>	游戏供应商的名称。“Sunbet”	类型：字符串 最大长度：50
<b>gameprovidercode</b>	游戏供应商代码，如“SB”。 请参阅附录K - 游戏供应商列表。	类型：字符串 最大长度：50字符。
<b>gamename</b>	游戏名称，如：“Live Dealer Baccarat”。	类型：字符串 最大长度：50字符。
<b>gameid</b>	游戏供应商的唯一游戏标识码。这是游戏类别标识码(如Blackjack)，非回合代码。	类型：字符串 最大长度：50字符。
<b>platformtype</b>	游戏平台的类型。可为下列其中之一： 请参阅 <a href="#">附录L - 平台类型</a> 的所有值。	类型：字符串 最大长度：50字符。
<b>ipaddress</b>	游戏供应商报告的玩家之IP地址。	类型：字符串 最大长度：15
<b>desc</b>	交易的描述。	类型：字符串 最大长度：255字符。
<b>playertype</b>	用来辨识是否为测试玩家或真实玩家。请参阅下面数值的意义：  1=真实玩家 2=运营商测试玩家	类型：整数

返回格式：csv。

### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

### Calls范例：

#### 请求(Request)

#### 调用：

```
http://<TGPIIntegrationAPIServer>api/history/bettransaction?userid=fa39953d&startdate=2015-08-14T07:07:31%2B00:00&enddate=2014-08-15T07:07:31%2B00:00
```

#### 响应(Response)

#### 请求成功的响应结果：

```
txid,ugstxid,timestamp,userid,username,amt,postbal,cur,roundid,roundstatus,actiontype,gameprovider,gameprovidercode,gamename,gameid,platformtype, ipaddress,playertype
6063c527,56677849-fb7b-49b8-ad1b-b393c54c01cf,2015-08-14T08:07:31+00:00,fa39953d,dave,-20.00,380.00,RMB,39390,Closed,Bet,Sunbet,SB, Live Dealer
Baccarat,Baccarat,Desktop,192.168.0.1
6063c528,2014-08-14T08:08:31+00:00,fa39953d,dave,10.00,400.00,RMB,39390,Closed,Win,Sunbet,Live Dealer Baccarat,Baccarat ,Mobile,192.168.0.1,1
```

### 7.6.3 游戏历史 (Game History)

此方法用来取得玩家的游戏历史。

**HTTP方法：**

GET

**内容类型(必需)：**

text/csv

**URL方法：**

`http://<TGPIntegrationAPIServer>/api/history/game`

**方法参数：**

参数	定义	必须有?	类型
<b>userid</b>	玩家唯一的标识代码。 ex: fc533ace-bb15 若未指定userid，所有日期区间内的记录都会被返回。	否	类型：字符串 最大长度：50字符。
<b>startdate</b>	游戏历史的开始日期(回合结束)。 'enddate'不得超过'startdate' 48小时。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T00:00:00%2B00:00
<b>enddate</b>	唯一的的游戏历史结束日期(回合结束)。 'enddate'不得超过'startdate' 48小时。 注意：这是唯一的，因此若要在2014-09-25 08:00这时间点取得一小时的数据，'startdate'和'enddate'应该分别是 2014-09-25T08:00:00%2B00:00和 2014-09-25T09:00:00%2B00:00。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T01:00:00%2B00:00
<b>includetestplayers</b>	是否包含测试玩家； 默认值为false。	否	类型：逻辑  eg. true

返回：

参数	定义	类型
<b>roundstart</b>	回合开始的时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
<b>roundend</b>	回合结束的时间，包括时区偏移量。若回合尚未结束，值为null。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
<b>roundid</b>	游戏交易执行回合(round)时的游戏供应商标识码。所有的交易都以回合作群组。	类型：字符串 最大长度：50
<b>ugsroundid</b>	游戏交易执行回合(round)时的UGS内部唯一的标识码。所有的交易都以回合作群组。	全局唯一标识符 最大长度：36字符。
<b>roundstatus</b>	报告回合时的状态。可为下列其中之一： Open Closed	类型：字符串 最大长度：10字符。
<b>userid</b>	玩家唯一的标识码。	类型：字符串 最大长度：50字符。
<b>username</b>	玩家的显示名称。	类型：字符串 最大长度：50字符。
<b>riskamt</b>	回合中所有投注的总投注金额。	类型：小数点 精确度：2位数。
<b>winamt</b>	回合中赢的金额，包含所有投注的总金额。 (注意：此字段不是赢的净额；Net win = winamt-riskamt)	类型：小数点 精确度：2位数。
<b>beforebal</b>	回合第一笔交易前玩家的余额。	类型：小数点 精确度：2位数。
<b>postbal</b>	回合最后一笔交易后玩家的余额。  请注意：因为可能同时会有多个游戏在玩，'beforebal'和'postbal'的差额不一定等于赢的净额。	类型：小数点 精确度：2位数。
<b>cur</b>	玩家用于TGP系统的货币(ISO 3字符代码)。	类型：字符串(ISO 3字符代码)

#### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

		长度：3字符。
<b>gameprovider</b>	游戏供应商的名称。“Sunbet”	类型：字符串 最大长度：50
<b>gameprovidercode</b>	游戏供应商代码，如“SB”。 请参阅附录K - 游戏供应商列表。	类型：字符串 最大长度：50字符。
<b>gamename</b>	游戏名称，如：“Live Dealer Baccarat”。	类型：字符串 最大长度：50字符。
<b>gameid</b>	游戏供应商的唯一游戏标识码。这是游戏类别标识码(如 Blackjack=13)，非回合代码。	类型：字符串 最大长度：50字符。
<b>platformtype</b>	游戏平台的类型。可为下列其中之一： 请参阅 <a href="#">附录L - 平台类型</a> 的所有值。	类型：字符串 最大长度：50字符。
<b>ipaddress</b>	游戏供应商报告的玩家之IP地址。	类型：字符串 最大长度：15
<b>playertype</b>	用来辨识是否为测试玩家或真实玩家。请参阅下面数值的意义：  1=真实玩家 2=运营商测试玩家	类型：整数

响应格式：csv。

#### 错误参数：

参数	定义	必须有?	类型
<b>err</b>	错误代码。	否	类型：整数
<b>errdesc</b>	以英文描述的错误说明。	否	类型：字符串



## Calls范例 :

### 请求(Request)

#### 调用 :

```
http://<TGPIIntegrationAPIServer>api/history/game?userid=fa39953d&startdate=2015-08-14T07:07:31%2B00:00&enddate=2014-08-15T07:07:31%2B00:00
```

### 响应(Response)

#### 请求成功的响应结果 :

```
roundstart,roundend,roundid,ugroundid,roundstatus,userid,username,playertype,riskamt,winamt,beforebal,postbal,cur,gameprovider,gameprovidercode,gamename,gameid,platformtype,ipaddress,playertype
2016-09-23T20:35:40+08:00,2016-09-23T20:35:48+08:00,876861474634024363,Closed,01_REAL_rmb_en-us,RP01_rmb_en-us,1,9.89,9.89,98964.44,98964.44,RMB,GG Fishing,GG,Fishing King,75608,Desktop,118.163.83.109,1
2016-09-24T03:49:57+08:00,2016-09-24T03:52:00+08:00,127666075400,56677849-fb7b-49b8-ad1b-b393c54c01cf,Closed,01_REAL_rmb_en-us,RP01_rmb_en-us,1,10.00,0.00,98964.44,98954.44,RMB,Globalbet,Virtual sports,1,Desktop,184.70.137.34 ,1
2016-09-24T09:13:51+08:00,2016-09-24T09:13:57+08:00,5198540,Closed,02_REAL_rmb_en-us,RP02_rmb_en-us,1,100.00,0.00,5015549.01,5015449.01, RMB,Laxino,Vegas Blackjack,1018,Desktop,202.175.186.234,1
2016-09-24T09:14:49+08:00,2016-09-24T09:14:54+08:00,5198542,Closed,02_REAL_rmb_en-us,RP02_rmb_en-us,1,1.50,0.25,5015449.01,5015447.76,RMB,Laxino,Great Winds Fortune,203001031,Desktop,202.175.186.234,1
```

### 7.6.4 投注历史 (Bet History)

投注历史方法是用来获取指定的日期/时间范围内的投注记录。建议运营商使用此方法来取得投注数据。当游戏回合的状态改变时(例如由开启变成关闭)，这回合的最后更新时间就会变更，你可使用这方法得到'startdate'和'enddate'参数所指定这段时间内的投注纪录。

此方法预设为返回已确定的和未确定的投注。若参数'issettled'设为True，则此方法只会回传已确定的投注。

#### HTTP方法：

GET

#### 内容类型(必需)：

text/csv

#### URL方法：

http://<TGPIntegrationAPIServer>/api/history/bets

#### 方法参数：

参数	定义	必须有?	类型
userid	玩家唯一的辨识代码。 ex: fc533ace-bb15 若未指定userid，所有日期区间内的记录都会被返回。	否	类型：字符串 最大长度：50字符。
startdate	游戏历史的开始日期(相较于一回合最后更新的时间)。 'enddate'不得超过'startdate' 48小时。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T00:00:00%2B00:00
enddate	唯一的的游戏历史结束日期(回合结束)。 'enddate'不得超过'startdate' 48小时。  注意：这是唯一的，因此若要在2014-09-25 08:00这时间点取得一小时的数据，'startdate'和'enddate'应该分别是 2014-09-25T08:00:00%2B00:00和 2014-09-25T09:00:00%2B00:00。	是	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T01:01:00%2B00:00
includetestplayers	是否包含测试玩家； 默认值为false。	否	类型：逻辑  eg. true

issettled	True : 只返回已确定的投注。 False : 返回已确定和未确定的投注(默认值)。	否	类型：逻辑  eg. true
-----------	---	---	-----------------------

#### 备注：

- 不建议指定大范围的'startdate'和'enddate'，如12小时。若如此，TGP就会返回在那12个小时区间内被更新的投注记录，也就是说会花比较久的时间将这些数据传回来。
- 若要每30分钟调用一次这个方法，可用'startdate'和'enddate'创建一个30分钟的区间。如：  
第一次调用：startdate=2016-06-03T09:00:00%2B08:00，enddate=2016-06-03T09:30:00%2B08:00  
第二次调用：startdate=2016-06-03T09:30:00%2B08:00，enddate=2016-06-03T10:00:00%2B08:00

#### 返回：

参数	定义	类型
ugsbetid	TGP内部投注标识码。这是唯一键值，可用来辨识特定的投注。  <b>注记：</b> 此字段很重要，因为可用多个API调用取得相同的投注。 如：  1. 第一次调用返回ugsbetid=1；betid=XYZ； roundstatus=open 2. 第二次调用返回ugsbetid=1；betid=XYZ； roundstatus=open  上面提到的'betid'是一个外部的投注标识码。参阅下面'betid'参数的说明。	类型：字符串 最大长度：50字符。
txid	来自游戏供应商的交易或投注标识码。	类型：字符串 最大长度：50字符。
betid	外部的投注标识码(来自游戏供应商)。 不要使用此字段去辨识特定的投注记录，因为可能会有相同的投注标识码来自不同的游戏供应商。	类型：字符串 最大长度：64字符。
beton	记录在TGP服务器的投注时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
betclosedon	投注关闭的时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。

		eg. 2014-09-25T07:07:31+00:00
<b>betupdatedon</b>	投注更新的时间，包括时区偏移量。	类型：日期(ISO) 长度：25字符。  eg. 2014-09-25T07:07:31+00:00
<b>timestamp</b>	游戏供应商处理动作的时间戳。若游戏供应商未提供则使用UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	类型：日期(ISO) 长度：25字符。
<b>roundid</b>	游戏交易执行回合(round)时的游戏供应商标识码。所有的交易都以回合作群组。	类型：字符串 最大长度：50字符。
<b>roundstatus</b>	报告回合时的状态。可为下列其中之一： - Open - Closed	类型：字符串 最大长度：10字符。
<b>userid</b>	玩家的唯一标识码(来自游戏供应商)。	类型：字符串 最大长度：50字符。
<b>username</b>	玩家的显示名称。	类型：字符串 最大长度：50字符。
<b>riskamt</b>	投注的总金额。此值为负数。	类型：小数点 精确度：2位数。
<b>winamt</b>	投注赢的金额。此值为正数或0。	类型：小数点 精确度：2位数。
<b>winloss</b>	投注的净总金额(winamt+riskamt)。	类型：小数点 精确度：2位数。
<b>beforebal</b>	投注交易前玩家的余额。  注意：因为可能同时会有多个游戏在玩，'beforebal'和'postbal'的差额不一定等于赢的净额。	类型：小数点 精确度：2位数。
<b>postbal</b>	投注交易后玩家的余额。  注意：因为可能同时会有多个游戏在玩，'beforebal'和'postbal'的差额不一定等于赢的净额。	类型：小数点 精确度：2位数。
<b>cur</b>	玩家用于TGP系统的货币(ISO 3字符代码)。	类型：字符串 长度：3字符。
<b>gameprovider</b>	游戏供应商的名称，如：“Sunbet”	类型：字符串

		最大长度：50
<b>gameprovidercode</b>	游戏供应商代码，如"SB"。 请参阅附录K -游戏供应商列表。	类型：字符串 最大长度：50字符。
<b>gamename</b>	游戏名称，如：“Live Dealer Baccarat”。	类型：字符串 最大长度：50字符。
<b>gameid</b>	游戏供应商的唯一游戏标识码。这是游戏标识码(如Blackjack=13)，非回合代码。	类型：字符串 最大长度：50字符。
<b>platformtype</b>	游戏平台的类型。 请参阅 <a href="#">附录L -平台类型</a> 的所有值。	类型：字符串 最大长度：50字符。
<b>ipaddress</b>	游戏供应商报告的玩家之IP地址。	类型：字符串 最大长度：15
<b>bettype</b>	游戏动作种类，可为下列其中之一： <ul style="list-style-type: none"> <li>- PlaceBet</li> <li>- WinBet</li> <li>- AdjustBet</li> <li>- CancelTransaction</li> <li>- LoseBet</li> <li>- FreeBet</li> </ul> 注意：目前这个值永远是“PlaceBet”，因为这个记录是'winbet'和'placebet'或'losebet'和'placebet'的合并结果。	类型：字符串 最大长度：30字符。
<b>playtype</b>	游戏供应商所定义的游戏类别。	类型：字符串 最大长度：50字符。  如： BaBank、BaDeal, DtTigr、DtDrag等。
<b>playertype</b>	用来辨识是否为测试玩家或真实玩家。请参阅下面数值的意义：  1=真实玩家 2=运营商测试玩家	类型：整数

响应格式：csv。

### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

### Calls范例：

#### 请求(Request)

#### 调用：

http://<TGPIntegrationAPIServer>/api/history/bets?startdate=2015-10-30T16%3A07%3A54%2B08%3A00&enddate=2015-10-30T16%3A08%3A54%2B08%3A00

#### 响应(Response)

#### 请求成功的响应结果：

```
ugsbetid,txid,betid,beton,betclosedon,betupdatedon,timestamp,roundid,roundstatus,useri
d,username,riskamt,winamt,winloss,beforebal,postbal,cur
,gameprovider,gameprovidercode,gamename,gameid,platformtype,ipaddress,bettype,playtype
,playertype
0577e07e-b619-4e12-a17a-56b438d82c51,548710,000298086,2015-10-30T14:44:55+08:00,2015-1
0-
30T14:45:16+08:00,2015-10-30T14:44:55+08:00,2015-10-30T06:45:51+08:00,4503567,Closed,a
ssjw888,assjw888,-40.00,80.00,40.00,128.00,168.00,RMB,Sunbet,SB,Dragon
Tiger,35,Desktop,119.142.42.61,PlaceBet,DtTig,1
```

## 7.6.5 来自游戏供应商的游戏历史

游戏供应商提供的游戏历史，也可通过TGP Operator API提供。

请注意，与前面节所述的History API方法不同，此API返回游戏供应商API返回的数据。这代表，如果游戏供应商程序返回HTML或XML，此API也将返回HTML或XML数据。这也意味着你作为客户端，需要从不同的游戏供应商端处理不同的格式。

### HTTP方法：

GET

### URL方法：

`http://<TGPIntegrationAPIServer>/api/history/providers/{gpcode}/rounds/{roundid}/users/{userid}?platformtype={platformtype}`

### 方法参数：

参数	定义	必须有?	类型
userid	玩家唯一的辨识代码。 ex: fc533ace-bb15	是	类型：字符串 最大长度：50
gpcode	游戏供应商唯一代码。 请参阅游戏供应商代码列表。	是	类型：字符串 最大长度：50
roundid	游戏交易执行回合(round)时的游戏供应商辨识码。所有的交易都以回合作群组。	是	类型：字符串 最大长度：50字符。
platformtype	启动游戏的平台。 请参阅 <a href="#">附录L - 平台类型</a> 的所有值。	是	类型：数量 默认值：0 (未指定桌面计算机)

## 返回：

从这个API返回的数据将会取决于游戏供应商的要求而不同。如：如果您正在请求Red Tiger的游戏历史。返回格式将是原始的HTML数据。

请参阅下表，返回内容的示例：

Game Provider	Contet-Type header	Body Example
Red Tiger	text/html	<pre> &lt;html&gt;&lt;head&gt; &lt;meta http-equiv="Content-Type" content="text/html; charset=utf-8"&gt; .....  &lt;div&gt;   &lt;h3&gt;Game Details&lt;/h3&gt;   &lt;table class="details" cellpadding="0" cellspacing="0"&gt;     &lt;tbody&gt;&lt;tr&gt;       &lt;td&gt; Game ID:&lt;/td&gt;       &lt;td&gt;LuckyWizard&lt;/td&gt;     &lt;/tr&gt;   &lt;/tbody&gt; &lt;/table&gt; ..... </pre>
Laxino	application/json	<pre> {   "url": "&lt;string&gt;" } </pre>
Sunbet	application/json	<pre> {   "return_code": 0,   "return_msg": "",   "extras": {},   "data": [     {       "tran_date": "2015-03-25T01:55:16.777",       "draw_no": 78198,       "instrument_id": 1,       "game_type": "Baccarat",       "game_result": "Banker [Banker:♣10-♣9       Player:♦6-♦4]",       "banker_result": "♣10-♣9",       "player_result": "♦6-♦4 ",       "status": "Normal",       "seat_results": null     }   ] } </pre>



<b>GoldDeluxe</b>	application/json	<pre>{   "GameInfo": {     "Product": "Baccarat",     "TableID": "B1",     "DateTime": "11/02/2016 14:34:35",     "GameID": "B11611022505",     "SubBetType": "Tie,Player",     "GameResult": "Player Hand:\r\n1: DIAMOND K, CLUB A, CLUB J \r\nBanker Hand:\r\n8: CLUB 9, SPADE 2, SPADE 7",     "WinningBet": "Banker, Big, Dragon, Player Odd, Player Small, Player Red, Player Diamond, Player 1, Banker Even, Banker Big, Banker Black, Banker Club, Banker 8",     "DealerID": "Dealer_6",     "Status": "Settled"   },   "ErrorDesc": null }</pre>
-------------------	------------------	---

### Calls范例 :

下面是一些API操作调用的示例。

#### 请求(Request)

POST /api/history/providers/**SB**/rounds/**ugs-20807857**/users/**gpiplayer**?platform=**0** HTTP/1.1  
 Content-Type: application/json  
 Authorization: Bearer pM8ZEKjbkn5kC7RqvdBn1gcZRfC8ccVxBbQmbIZYnVYeb4LirxfS9XARFvPJaoKjM

#### 响应(Response)

**Baccarat请求成功的响应结果 :**

```
{
  "return_code": 0,
  "return_msg": "",
  "extras": {},
  "data": [
    {
      "tran_date": "2015-03-25T01:55:16.777",
      "draw_no": 78198 ,
      "instrument_id": 1,
      "game_type": "Baccarat",
      "game_result": "Banker [Banker:♠10-♣9 Player:♦6-♦4]",
      "banker_result": "♠10-♣9" ,
      "player_result": "♦6-♦4",
      "status": "Normal",
      "seat_results": null
    }
  ]
}
```

```

    }
  ]
}

```

#### Sic Bo请求成功的响应结果：

```

{
  "return_code": 0,
  "return_msg": "",
  "extras": {},
  "data": [
    {
      "tran_date": "2015-03-25T01:20:51.753",
      "draw_no": 50877,
      "instrument_id": 20,
      "game_type": "Sicbo",
      "game_result": "3, 5, 4",
      "banker_result": null,
      "player_result": null,
      "status": "Normal",
      "seat_results": null
    }
  ]
}

```

#### 如果从供应商返回错误，则是失败的结果：

```

{
  "return_code": 1,
  "return_msg": "Round not exists",
  "extras": {},
  "data": null
}

```

#### 错误参数：

此API将会把错误以JSON数据返回。包含：Content-Type: application/json响应标头。

参数	定义	必须有?	类型
err	错误代码。	否	类型：整数
errdesc	以英文描述的错误说明。	否	类型：字符串

#### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

## 8. 单一电子钱包集成

[**注记：单一电子钱包API是测试版。使用此测试版之前，请向UGS开发人员确认。**]

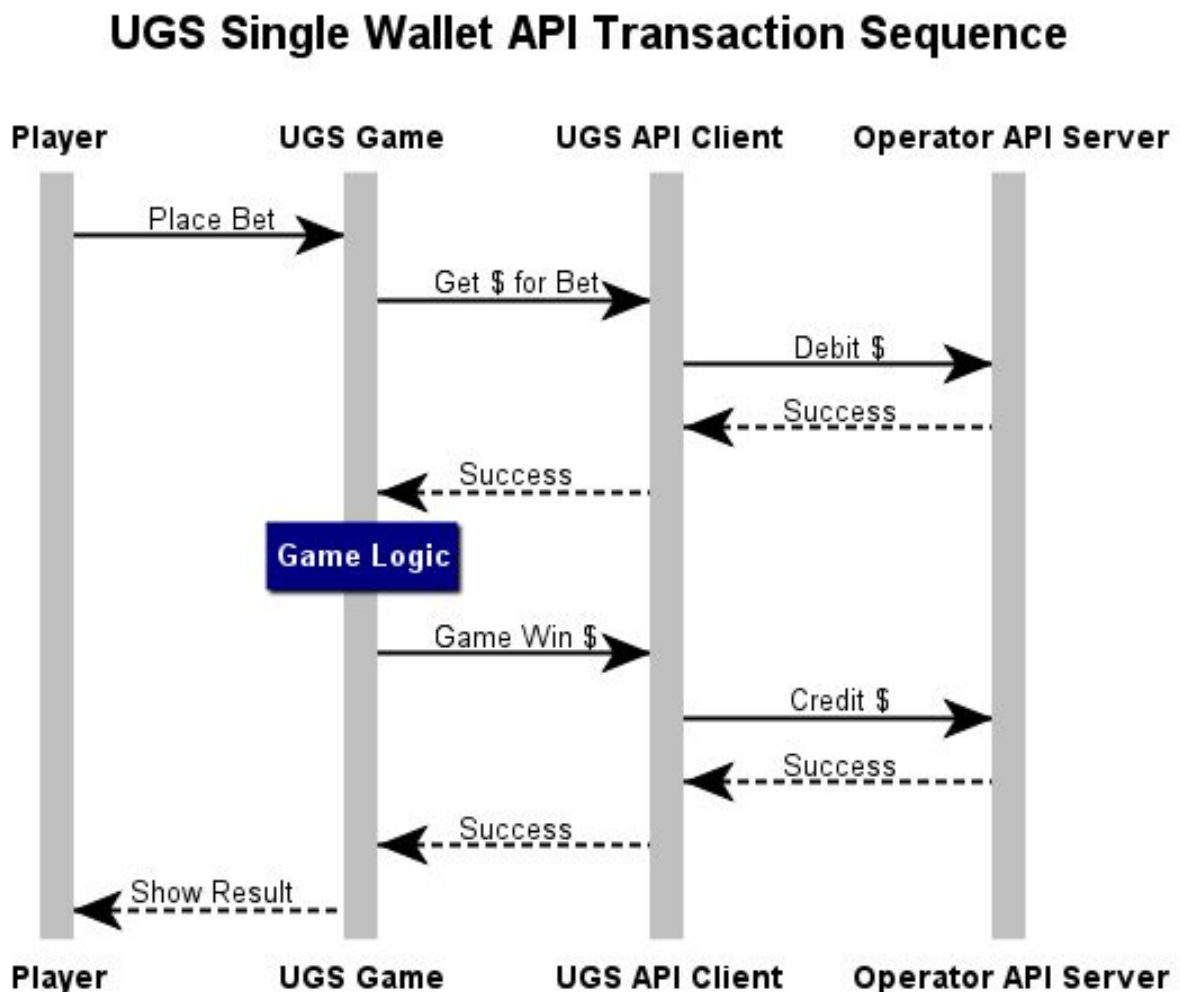
运营商可能会想要使用单一电子钱包的API，可让UGS系统将交易从游戏供应商传递给运营商，无需经过中间电子钱包。这亦称为"共同电子钱包"集成。

注意：此集成方法需要运营商提供此章节所描述的API方法。

此一作业称为"运营商单一电子钱包API服务器"，或简称为"运营商API服务器"。

调用这API的"UGS单一电子钱包客户端"被称为"UGS单一电子钱包API客户端"，或简称为"UGS API客户端"

### 8.1 顺序图



title UGS单一电子钱包API交易顺序

玩家-> UGS游戏：下注

UGS游戏-> UGS API客户端；取得投注的钱

UGS API客户端->运营商API服务器：扣钱(Debit)

运营商API服务器-> UGS API客户端：成功

UGS API客户端-> UGS游戏：成功

note over UGS游戏：游戏逻辑

UGS游戏-> UGS API客户端；游戏赢钱

UGS API客户端->运营商API服务器：加钱(Credit)

运营商API服务器-> UGS API客户端：成功

UGS API客户端-> UGS游戏：成功

UGS游戏->玩家：显示结果

## 8.2 客户端服务器授权

OAuth2授权必须由运营商API服务器执行，用来验证UGS API客户端已获授权与其通信。

为了实现OAuth 2.0标准，会依据一个共享的客户端标识码(Client ID)和客户端密钥(Client Secret)产生一个访问令牌(Access Token)，并在通信创建时提供给运营商。然后，所有的服务器API请求皆会将此访问令牌当作Authorization HTTP Header的Bearer值。

访问令牌的默认有效时间为1小时，一旦过其，必须要刷新才可继续执行API调用。

### HTTP方法：

POST

### HTTP header:

内容类型：application/x-www-form-urlencoded

### URL方法：

http://<IntegrationURL>/wallet/token

### 方法参数：

参数	定义	必须有?	类型
client_id	UGS的唯一标识码，由运营商在设定时提供。	是	类型：字符串 最大长度： 50字符。
client_secret	共享的密钥，由运营商在设定时提供。	是	类型：字符串 最大长度： 200
grant_type	永远是“client_credentials”。	是	类型：字符串
scope	永远是“wallet”。	是	类型：字符串

### 返回参数：

参数	定义	必须有?	类型
access_token	调用运营商API服务器的访问令牌。	是	类型：字符串 最大长度：255字符。
token_type	永远是“bearer”。	是	类型：字符串
expires_in	时间终止以秒计算，共3600秒(1小时)。	是	类型：整数
scope	永远是“wallet”。	是	类型：字符串

### 错误参数：

参数	定义	必须有?	类型
err	错误代码。	否	类型：字符串
err_description	以英文描述的错误说明。	否	类型：字符串

### Calls范例：

请求(Request)
<p>POST /wallet/token HTTP/1.1</p> <p>Content-Type: application/x-www-form-urlencoded</p> <p>grant_type=client_credentials&amp;scope=wallet&amp;client_id=Single-Wallet-Integration&amp;client_secret=1uuvtpXapvS6RwslyA2FgrZQA7V5EoPNT8djr0o8wqf1</p>
响应(Response)
<p>请求成功的响应结果：</p> <pre>{   "access_token": "JXhHKcdt3rELh0GqPY0PCrH4NgZwBxhxPsucdZuTanIAfZfOKYnmBHaY6atcks0R2",   "token_type": "Bearer",   "expires_in": 3600,   "scope": "wallet" }</pre>

### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

请求失败的响应结果：

```
{
  "error": "invalid_scope ",
  "error_description": "Invalid Scope"
}
```

欲知OAuth错误代码，请参阅[附录A](#)。

## 8.3 扣钱

一旦UGS API客户端通过验证后，就可对运营商API服务器执行扣钱的动作，将钱从玩家电子钱包提出。此方法会依据OAuth标准，要求在标头提供OAuth令牌。

注意，在此请求中可能会有多个交易(请参阅下面的'transactions'参数)，且当txtype=500 (placebet)时，'transactional'参数值是True，意即这些交易应该全部都成功或全部都失败。

### HTTP方法：

POST

### HTTP header:

Content-Type: application/json

Authorization: Bearer <[Access Token](#)>

### URL方法：

http://<IntegrationURL>/wallet/debit

### 方法参数：

参数	说明	类型
<b>testmode</b> (可选)	接受： delay- 10会模拟慢速请求(10秒)。 delay-5会模拟慢五秒的请求。 fail-500会模拟500错误回应。 emptyresponse会返回空的响应。	字符串
<b>transactional</b>	True表示此请求中所有的交易应该全部都成功或全部都失败。 False表示这些交易可各别成功。	逻辑
<b>transactions</b>	包含下列字段的对象数组容器。  注意：根据不同的游戏类型，可能有一个或多个交易在此数组中，如：一个玩家可以在同一回合的真人娱乐游戏中有多个投注。	数组
<b>userid</b>	玩家唯一的辨识代码。	字符串 最大长度： 50
<b>authtoken</b>	若玩家令牌存在。	字符串 最大长度： 2000字符。
<b>brandcode</b>	品牌唯一辨识码。	字符串 最大长度： 20

### TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB



amt	交易金额。必须大于0。值为0的交易，必须使用加钱操作。	小数点 精确度2位数。
cur	交易货币，使用品牌组态设定的货币代码。	字符串 长度：3到8字符。
ipaddress	玩家客户端IP地址。	字符串 最大长度：15
ptxid	特定动作的UGS平台标识码。不可重复。若送出重复的标识码，前一次结果会被重送，且会设有重复旗标。此与UGS的GameActionId一致。	全局唯一标识符 最大长度：36字符。
refptxid	若此参数被赋予一个值，它将会指向一个UGS平台交易标识码。 例如，当'txttype'的值是560（取消交易），'refptxid'的值就是取消动作的目标交易。  当txttype的值是510、520、530、540、或560时，此栏位可以有一个值。	全局唯一标识符 最大长度：36字符。
txtype	交易类别，列表于 <a href="#">附录F</a> 。	整数
timestamp	游戏供应商处理动作的时间戳。若游戏供应商未提供则使用UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	日期(ISO) 长度：25字符。
platformtype	玩家的平台。 请参阅 <a href="#">附录L - 平台类型</a> 的所有值。	类型：整数
gpcode	游戏供应商唯一代码。	字符串 最大长度：64字符。
gamecode	UGS游戏唯一代码。	字符串 最大长度：50字符。
gamename	UGS游戏的简称。	字符串 最大长度：50字符。
gametype	游戏种类(0=Slot；1=TableGame)，列表于 <a href="#">附录H</a> 。	整数
externalgameid	游戏供应商的游戏唯一标识码。	字符串 最大长度：50字符。
roundid	回合(round)或手牌(hand)或换牌(draw)的UGS唯一标识码。回合加入投注群组。	全局唯一标识符 最大长度：36字符。
externalroundid	回合或手牌或换牌的游戏供应商唯一标识码。	字符串 最大长度：50字符。

		符。
<b>betid</b>	投注的UGS标识码。	全局唯一标识符 最大长度： 36字符。
<b>externalbetid (可选)</b>	投注的游戏供应商标识码。并非一直都是由游戏供应商提供。	字符串 最大长度： 50字符。
<b>senton</b>	游戏供应商处理动作的时间戳。若游戏供应商未提供则使用UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	日期(ISO) 长度： 25字符。
<b>isclosinground</b>	旗标，显示此动作是否关闭回合。	逻辑 (true或false)
<b>ggr</b>	此事件所产生的游戏毛收入贡献。 此值可以为负。	小数点 精确度3位数。
<b>turnover</b>	此事件所产生的营业额贡献。 此值不为负数。	小数点 精确度3位数。
<b>unsettledbets</b>	此事件所产生的未确定投注贡献。这是个调整值，不是现行总和，且可以是负值。若不需要这个值，运营商可以将其忽略。	小数点 精确度3位数。
<b>walletcode (可选)</b>	选择性参数，用来指定使用运营商端哪一个电子钱包。	字符串 最大长度： 20
<b>bonustype (可选)</b>	起始此游戏动作的奖金类型，列表列于 <a href="#">附录G</a> 。	整数
<b>bonuscode (可选)</b>	起始此游戏动作的奖金奖励。	字符串 最大长度： 50字符。
<b>desc (可选)</b>	来自游戏供应商的扩充描述。投注种类可存储在此栏位中，如“BaBank”，“BaPlayer”...等。	字符串 最大长度： 2000字符。

## 返回参数：

参数	说明	类型
<b>err</b>	错误代码(900)。 若无错误就不会出现。  此顶级错误表示整个请求的错误，且所有的事情都失败，也就是说'transactions'参数(如下显示)应该是空的数组。 若有一些交易成功，一些失败，那些个别的错误应该在'transactions'节点中被返回，且此顶级错误不应出现。	整数
<b>errdesc</b>	以英文描述的错误说明。 若无错误就不会出现。	字符串
<b>transactions</b>	包含下列字段的对象数组容器。	数组
<b>txid</b>	唯一动作的运营商交易标识码。	字符串 最大长度：50字符。
<b>ptxid</b>	在请求中所提供唯一动作的UGS平台交易标识码。	全局唯一标识符 长度：36字符。
<b>bal</b>	交易完成后的玩家余额。	小数点 精确度：小数点两位
<b>cur</b>	玩家的货币，使用品牌组态设定的货币代码。	字符串 长度：3到8字符。
<b>dup</b>	重复旗标，显示是否为重送的结果。注记：若是重送的结果，'bal'字段应内含当前余额。	逻辑：(true或false)
<b>err</b>	错误代码。 若无错误就不会出现。	整数
<b>errdesc</b>	以英文描述的错误说明。 若无错误就不会出现。	字符串

## 调用范例：

### 请求(Request)

```
POST /wallet/debit HTTP/1.1
Content-Type: application/json
Authorization: Bearer pM8ZEKjbkn5kC7RqvdBn1gcZRfC8ccVxBbQmbIZYnVYeb4LirxfS9XARFvPJJa0KjM

{
  "transactional": "true",
  "transactions": [
    {
      "userid": "bob123",
      "authtoken": "j9rlVRbwRsKG0pE2l55fn8Cw5TF2guj81tSr...(trimmed)",
      "brandcode": "brandABC",
      "amt": 50,
      "cur": "RMB",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "refptxid": null,
      "txtype": 500,
      "timestamp": "2015-11-26T07:16:03+00:00",
      "platformtype": 0,
      "gpcode": "FC",
      "gamecode": "Fortune_God",
      "gamename": "Fortune God",
      "gametype": 0,
      "externalgameid": "FG-FLASH",
      "roundid": "56677849-fb7b-49b8-ad1b-b393c54c01cf",
      "externalroundid": "28849c84-fb7b-49b8-ad1b-b393c54c01cf",
      "betid": "28849c84-fb7b-49b8-ad1b-b393c54c01cf",
      "senton": "2015-11-26T07:16:01+08:00",
      "isclosinground": "false",
      "ggr": 0,
      "turnover": 50,
      "unsettledbets": 50,
      "walletcode": "LiveDealerWallet",
      "desc": ""
    }
  ]
}
```

## 响应(Response)

请求成功的响应结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "bal": 206.5,
      "cur": "RMB",
      "dup": false
    }
  ]
}
```

重复交易的成功结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "dup": true
    }
  ]
}
```

失败的结果：

```
{
  "err": 10,
  "errdesc": "Token has expired"
}
```

## 8.4 加钱(Credit)

一旦UGS API客户端通过验证后，就可对运营商API服务器执行加钱的动作。此方法会依据OAuth标准，要求在标头提供OAuth令牌。

### HTTP方法：

POST

### HTTP header:

Content-Type: application/json

Authorization: Bearer <[Access Token](#)>

### URL方法：

http://<IntegrationURL>/wallet/credit

### 方法参数：

参数	说明	类型
<b>testmode</b> (可选)	接受： delay- 10会模拟慢速请求(10秒)。 delay-5会模拟慢五秒的请求。 fail-500会模拟500错误回应。 emptyresponse会返回空的响应。	字符串
<b>platformtype</b>	玩家的平台。 请参阅 <a href="#">附录L-平台类型</a> 的所有值。	类型：整数
<b>transactions</b>	包含下列字段的对象数组容器。  注意：可能有一个或多个交易在此数组中，这些交易可能成功也可能失败(详情请参阅下面的返回参数)。	数组
<b>userid</b>	玩家唯一的辨识代码。	字符串 最大长度：50
<b>authtoken</b>	若玩家令牌存在。	字符串 最大长度：2000字符。
<b>brandcode</b>	品牌唯一辨识码。	字符串 最大长度：20
<b>amt</b>	交易金额。可以是大于或等于0。	小数点 精确度2位数。

<b>cur</b>	交易货币，使用品牌组态设定的货币代码。	字符串 长度： 3到8字符。
<b>ipaddress</b>	玩家客户端IP地址。	字符串 最大长度： 15
<b>ptxid</b>	特定动作的UGS平台标识码。不可重复。若送出重复的标识码，前一次结果会被重送，且会设有重复旗标。此与UGS的GameActionId一致。	全局唯一标识符 最大长度： 36字符。
<b>refptxid</b>	若此参数被赋予一个值，它将会指向一个UGS平台交易标识码。例如，当'txttype'的值是560（取消交易），'refptxid'的值就是取消动作的目标交易。  当txttype的值是510、520、530、540、或560时，此栏位可以有一个值。	全局唯一标识符 最大长度： 36字符。
<b>txtype</b>	交易类别，列表于 <a href="#">附录F</a> 。	整数
<b>timestamp</b>	游戏供应商处理动作的时间戳。若游戏供应商未提供则使用UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	日期(ISO) 长度： 25字符。
<b>platformtype</b>	玩家的平台。 请参阅 <a href="#">附录L -平台类型</a> 的所有值。	类型：整数
<b>gpcode</b>	游戏供应商唯一代码。	字符串 最大长度： 50字符。
<b>gamecode</b>	UGS游戏唯一代码。	字符串 最大长度： 50字符。
<b>gamename</b>	UGS游戏的简称。	字符串 最大长度： 50字符。
<b>gametype</b>	游戏种类(0=Slot；1=TableGame)，列表于 <a href="#">附录H</a> 。	整数
<b>externalgameid</b>	游戏供应商的游戏唯一标识码。	字符串 最大长度： 50字符。
<b>roundid</b>	回合(round)或手牌(hand)或换牌(draw)的UGS唯一标识码。回合加入投注群组。	全局唯一标识符 最大长度： 36字符。
<b>externalroundid</b>	回合或手牌或换牌的游戏供应商唯一标识码。	字符串 最大长度： 50字符。
<b>betid</b>	投注的UGS标识码。	全局唯一标识符

		最大长度： 36 字符。
<b>externalbetid</b> (可选)	投注的游戏供应商标识码。并非一直都是由游戏供应商提供。	字符串 最大长度： 50 字符。
<b>senton</b>	游戏供应商处理动作的时间戳。若游戏供应商未提供则使用UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	日期(ISO) 长度： 25 字符。
<b>isclosingground</b>	旗标，显示此动作是否关闭回合。	逻辑 (true或false)
<b>ggr</b>	此事件所产生的游戏毛收入贡献。 此值可以为负。	小数点 精确度3位数。
<b>turnover</b>	此事件所产生的营业额贡献。 此值不为负数。	小数点 精确度3位数。
<b>unsettledbets</b>	此事件所产生的未确定投注贡献。这是个调整值，不是现行总和，且可以是负值。若不需要这个值，运营商可以将其忽略。	小数点 精确度3位数。
<b>walletcode</b> (可选)	选择性参数，用来指定使用运营商端哪一个电子钱包。	字符串 最大长度： 20
<b>bonustype</b> (可选)	起始此游戏动作的奖金类型，列表列 <a href="#">附录G</a> 。	整数
<b>bonuscode</b> (可选)	起始此游戏动作的奖金奖励。	字符串 最大长度： 50 字符。
<b>desc</b> (可选)	来自游戏供应商的扩充描述。	字符串 最大长度： 2000 字符。



## 返回参数：

参数	说明	类型
<b>err</b>	错误代码(900)。 若无错误就不会出现。  此顶级错误表示整个请求的错误，且所有的事情都失败，也就是说'transactions'参数(如下显示)应该是空的数组。 若有一些交易成功，一些失败，那些个别的错误应该在'transactions'节点中被返回，且此顶级错误不应出现。	整数
<b>errdesc</b>	以英文描述的错误说明。 若无错误就不会出现。	字符串
<b>transactions</b>	包含下列字段的对象数组容器。 每个交易的结果会以此数组的一个元素来描述，他们可能成功也可能失败。	数组
<b>txid</b>	此动作的运营商交易标识码。	字符串 最大长度：50字符。
<b>ptxid</b>	在请求中所提供之动作的UGS平台交易标识码。	全局唯一标识符 长度：36字符。
<b>bal</b>	交易完成后的玩家余额。	小数点 精确度：小数点两位
<b>cur</b>	玩家的货币。	字符串 长度：3到8字符。
<b>dup</b>	重复旗标，显示是否为重送的结果。注记：若是重送的结果，'bal'字段应内含当前余额。	逻辑：(true或false)
<b>err</b>	错误代码。 若无错误就不会出现。	整数
<b>errdesc</b>	以英文描述的错误说明。 若无错误就不会出现。	字符串

## 调用范例：

### 请求(Request)

POST /wallet/credit

#### Header:

Content-Type: application/json

Authorization: Bearer pM8ZEKjbkn5kC7RqvdbN1gcZRfC8ccVxBbQmbIZYnVVEb4LirxfS9XARFvPJaoKjM

#### Content:

```
{
  "transactions": [
    {
      "userid": "bob123",
      "authtoken": "j9r1VRbwRsKG0pE2l55fn8Cw5TF2guj81tSr...(trimmed)",
      "brandcode": "brandABC",
      "amt": 50,
      "cur": "RMB",
      "ipaddress": "127.0.0.1",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "refptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a123",
      "txtype": 510,
      "timestamp": "2014-09-25T07:07:31+00:00",
      "platformtype": 0,
      "gpcode": "FC",
      "gamecode": "Fortune_God",
      "gamename": "Fortune God",
      "gametype": 0,
      "externalgameid": "FG-FLASH",
      "roundid": "56677849-fb7b-49b8-ad1b-b393c54c01cf",
      "externalroundid": "28849c84-fb7b-49b8-ad1b-b393c54c01cf",
      "betid": "28849c84-fb7b-49b8-ad1b-b393c54c01cf",
      "senton": "2014-09-25T07:07:30+00:00",
      "isclosinground": "false",
      "ggr": 50,
      "turnover": 50,
      "unsettledbets": 0,
      "walletcode": "LiveDealerWallet"
    }
  ]
}
```

## 响应(Response)

请求成功的响应结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "bal": 206.5,
      "cur": "RMB",
      "dup": false
    }
  ]
}
```

重复交易的成功结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "dup": true
    }
  ]
}
```

个别失败交易的成功结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "bal": 206.5,
      "cur": "RMB",
      "dup": false
    },
    {
      "txid": "662675",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a728",
      "err": 600,
    }
  ]
}
```

```

    "errdesc": "Transaction not exists",
  }
]
}

```

**失败的结果(在顶层节点中的返回错误)：**

```

{
  "err": 10,
  "errdesc": "Token has expired"
}

```

## 8.5取消(Cancel)

一旦UGS API客户端通过验证后，就可对运营商API服务器执行取消的动作。此方法会依据OAuth标准，要求在标头提供OAuth令牌。

此取消动作可用来取消前次尚未被取消的扣钱交易。

**注记：** *Cancel* 操作不应与游戏供应商在Debit和Credit操作时所启动的 *CancelTransaction* 混淆 (txtype=560)。 *Cancel* 指令是由UGS所启动，仅用于取消txtype=500失败的Debit操作(即PlaceBet交易)。换句话说，Cancel Transaction (txtype=560)是游戏供应商所启动的一个请求，可用于取消先前成功的下注或结算交易。失败的Credit操作不会使用Cancel指令，但会不断尝试执行直到成功为止。

欲知详细说明，请参阅 [8.9节-错误和重试处理](#)。

## HTTP方法：

POST

## HTTP header:

Content-Type: application/json

Authorization: Bearer <[Access Token](#)>

## URL方法：

http://<IntegrationURL>/wallet/cancel

## 方法参数：

参数	说明	类型
<b>testmode</b> (可选)	接受： delay- 10会模拟慢速请求(10秒)。 delay-5会模拟慢五秒的请求。 fail-500会模拟500错误响应。 emptyresponse会返回空的响应。	字符串
<b>transactions</b>	包含下列字段的对象数组容器。	数组
<b>userid</b>	玩家唯一的辨识代码。	字符串 最大长度： 50
<b>authtoken</b>	若玩家令牌存在。	字符串 最大长度： 2000字符。
<b>brandcode</b>	品牌唯一辨识码。	字符串 最大长度： 20
<b>refptxid</b>	被取消动作的UGS平台交易辨识码。若交易不存在，就会产生错误代码600。若交易已被取消，就会产生错误代码610。(请参阅 <a href="#">附录J</a> )	全局唯一标识符 最大长度： 36字符。
<b>timestamp</b>	UGS处理动作的时间戳。  eg. 2014-09-25T07:07:31+00:00	日期(ISO) 长度： 25字符。
<b>walletcode</b> (可选)	选择性参数，用来指定使用运营商端哪一个电子钱包。	字符串 最大长度： 20

## TGP 运营商 API 集成指南

TGP Asia Limited, 2nd Floor, Athol House, 21a – 23 Athol Street, Douglas, Isle of Man, IM1 1LB

## 返回参数：

参数	说明	类型
err	错误代码(900)。 若无错误就不会出现。  此顶级错误表示整个请求的错误，且所有的事情都失败，也就是说'transactions'参数(如下显示)应该是空的数组。 若有一些交易成功，一些失败，那些个别的错误应该在'transactions'节点中被返回，且此顶级错误不应出现。	整数
errdesc	以英文描述的错误说明。 若无错误就不会出现。	字符串
transactions	包含下列字段的对象数组容器。	数组
txid	此动作的运营商交易标识码。	字符串 最大长度：50字符。
ptxid	在请求中所提供之动作的UGS平台交易标识码。	全局唯一标识符 长度：36字符。
bal	交易完成后的玩家余额。	小数点 精确度：小数点两位
cur	玩家的货币。	字符串 长度：3到8字符。
dup	重复旗标，显示是否为重送的结果。注记：若是重送的结果，'bal'字段应内含当前余额。	逻辑：(true或false)
err	错误代码。 若无错误就不会出现。	整数
errdesc	以英文描述的错误说明。 若无错误就不会出现。	字符串

## 调用范例：

### 请求(Request)

POST /wallet/cancel HTTP/1.1  
 Content-Type: application/json  
 Authorization: Bearer pM8ZEKjbkn5kC7RqvdBn1gcZRfC8ccVxBbQmbIZYnVYEb4LirxfS9XARFvPJJaOKjM

```
{
  "transactions": [
    {
      "userid": "bob123",
      "brandcode": "brandABC",
      "refptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "timestamp": "2015-11-26T07:16:03+00:00",
      "walletcode": "LiveDealerWallet",
      "desc": "Cancellation due to network error"
    }
  ]
}
```

### 响应(Response)

请求成功的响应结果：

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727 ",
      "bal": 206.5,
      "cur": "RMB",
      "dup": false
    }
  ]
}
```

请求失败的响应结果：

```
{
  "err": 10,
  "errdesc": "Token has expired"
}
```



## 8.6取得余额(GetBalance)

一旦UGS API客户端通过验证后，就可在运营商API服务器上执行取得余额的动作，返回一个或多个玩家电子钱包的余额。此方法会依据OAuth标准，要求在标头提供OAuth令牌。

### HTTP方法：

POST

### HTTP header:

Content-Type: application/json

Authorization: Bearer <[Access Token](#)>

### URL方法：

http://<IntegrationURL>/wallet/balance

## 方法参数：

参数	说明	类型
<b>testmode</b> (可选)	接受： delay- 10会模拟慢速请求(10秒)。 delay-5会模拟慢五秒的请求。 fail-500会模拟500错误响应。 emptyresponse会返回空的响应。 setbalance:1000会在测试封包仅存1000。	字符串
<b>users</b>	包含下列字段的对象数组容器。  可能有一个或多个用户在此数组中，如果运营商没有找到userid，那么直接忽略它，不需要再返回该玩家的信息(且该玩家也没有错误发生)。	数组
<b>authtoken</b>	若玩家令牌存在。 注记：一些游戏供应商没有提供玩家令牌，当他们将批次请求发送到UGS要取得多个玩家的余额量，所以此字段可能为空值。	字符串 最大长度：2000字符。
<b>userid</b>	来自品牌的玩家唯一标识代码。，以'brandcode'参数指定。	字符串 最大长度：50字符。
<b>brandcode</b>	品牌唯一标识码。	字符串 最大长度：20
<b>lang</b> (可选)	玩家所使用的语言代码。若语言代码未显示或为空值，或在客户端未支持，请使用玩家的默认语言。 请参阅 <a href="#">附录D</a> 语言代码的列表。	类型：字符串 长度：5字符。
<b>cur</b>	交易货币，使用品牌组态设定的货币代码。	字符串 长度：3到8字符。
<b>walletcode</b> (可选)	选择性参数，用来指定使用运营商端哪一个电子钱包。  注记：若未显示，请返回所指定'userid'所有的钱包。	字符串 最大长度：20

## 返回参数：

参数	说明	类型
<b>err</b>	错误代码。 若无错误就不会出现。	整数
<b>errdesc</b>	以英文描述的错误说明。 若无错误就不会出现。	字符串
<b>users</b>	包含下列字段的对象数组容器。  注记：如果运营商端不存在该userid，那么直接忽略该玩家(不需要把该玩家放在此数组中)。	数组
<b>用户数组的子对象。</b>		
<b>userid</b>	品牌的唯一玩家标识码。该用户的各项钱包存储在下面'wallets'的数组中。	字符串 最大长度：50字符。
<b>err</b>	返回此用户余额失败时的错误代码。 若无错误就不会出现。	整数
<b>errdesc</b>	以英文描述的错误说明。若无错误就不会出现。	字符串
<b>wallets</b>	包含一个或多个钱包的对象数组。	数组
<b>钱包数组的子对象。</b>		
<b>code</b>	钱包代码。	字符串 最大长度：20字符。
<b>bal</b>	玩家的余额量。	类型：小数点 精确度：2位数。
<b>cur</b>	玩家的货币。	字符串 长度：3到8字符。
<b>name (可选)</b>	钱包名称的本地化字符串。	字符串 最大长度：50字符。
<b>desc (可选)</b>	钱包描述的本地化字符串。	字符串 最大长度：256字符。

## 示例：

### 请求(Request)

```
POST /wallet/balance HTTP/1.1
Content-Type: application/json
Authorization: Bearer pM8ZEKjbnk5kC7RqvdBn1gcZRfC8ccVxBbQmbIZYnVYEb4LirxfS9XARFvPJJa0KjM

{
  "users": [
    {
      "userid": "bob123",
      "brandcode": "brandABC",
      "cur": "RMB",
      "lang": "zh-CN",
      "walletcode": "LiveDealerWallet"
    },
    {
      "userid": "mary345",
      "brandcode": "brandABC",
      "cur": "RMB",
      "lang": "zh-CN",
      "walletcode": "SlotWallet"
    }
  ]
}
```

### 响应(Response)

#### Successful result:

```
{
  "users": [{
    "userid": "bob123",
    "wallets": [{
      "code": "LiveDealerWallet",
      "bal": 206.5,
      "cur": "RMB",
      "name": "Live Dealer wallet",
      "desc": "真人荷官钱包"
    }],
    {
```

```

        "code": "SlotWallet",
        "bal": 100,
        "cur": "RMB",
        "name": "Slot wallet",
        "desc": "真人荷官钱包"
    ]
}, {
    "userid": "mary345",
    "wallets": [{
        "code": "SlotWallet",
        "bal": 5506.5,
        "cur": "RMB",
        "name": "Slot wallet",
        "desc": "老虎机钱包"
    }]
}]
}

```

整个请求失败的失败结果：

```

{
  "err": 10,
  "errdesc": "Token has expired"
}

```

个别玩家错误的失败结果：

```

{
  "users": [
    {
      "userid": "bob123",
      "wallets": [
        {
          "code": "LiveDealerWallet",
          "bal": 206.5,
          "cur": "RMB",
        }
      ]
    },
    {
      "userid": "abc123",
      "err": 50,
      "errdesc": "Currency Mismatch"
    }
  ]
}

```

## 8.7 交易群组

Credit、Debit和Cancel指令操作可包含多重交易。这是为了便于批量处理，亦指定必须连锁处理的交易集。

下表指示每个指令操作所支持的群组特性。

指令操作	属于交易性质？	多重玩家	多重交易	多重回合
扣钱(Debit)	是	是	是	否
加钱(Credit)	否	是	是	否
取消(Cancel)	否	否	是	否

### 栏位定义

<b>属于交易性质？</b>	该指令操作是否支持“transactional”旗标，表示这些交易是否必须以群组型态表示成功或失败。
<b>多重玩家</b>	该指令操作是否在相同请求下支持多重玩家？  注记：支持多重玩家并不表示一定会有多重玩家。例如：txtype=500 (投注)的Debit操作就不会有多重玩家。
<b>多重交易</b>	该指令操作是否在相同请求下支持多重交易？
<b>多重回合</b>	该指令操作是否在相同请求下针对不同的游戏回合支持交易处理请求？

## 8.8 返回错误响应

当没有错误发生时，'err'和'errdesc'属性就不会出现在响应信息内容中，或'err'的值应为0。

### 整体操作失败

当响应信息内容有'err'和'errdesc'在顶层节点时，UGS会视其为失败的操作，亦即在这指令操作中的所有动作/该笔交易全部皆失败。

Example:

```
{
  "err": 10,
  "errdesc": "Token has expired"
}
```

### 部分失败

当指令操作包含部分成功和失败结果时，顶层节点不应包含'err'和'errdesc'的属性。相反地，它会在次节点中描述各别错误。

示例1:包含两个成功和一个失败交易的钱包/加钱响应。注意，重复的交易不是错误。

```
{
  "transactions": [
    {
      "txid": "662674",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a727",
      "bal": 206.5,
      "cur": "RMB",
      "dup": false
    },
    {
      "txid": "662675",
      "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a728",
      "bal": 220.0,
      "cur": "RMB",
      "dup": true
    },
    {
      "txid": "662675",
```

```

        "ptxid": "c6c2fcf9-4685-4a91-8f57-e5185655a729",
        "err": 600,
        "errdesc": "transaction not exist. refptxid=12345"
    }
]
}

```

示例2: 包含一个成功钱包余额(userid:'bob123')和一个错误(userid:'abd123')的钱包/余额响应。

```

{
  "users": [
    {
      "userid": "bob123",
      "wallets": [
        {
          "code": "LiveDealerWallet",
          "bal": 206.5,
          "cur": "RMB",
        }
      ]
    },
    {
      "userid": "abc123",
      "err": 50,
      "errdesc": "Currency Mismatch"
    }
  ]
}

```



## 8.9 错误和重试处理(Errors and Retry Handling)

UGS单一电子钱包API客户端可弹性处理运营商API服务器的错误并恢复运作。

本章节描述UGS如何处理错误和指令重试。

### 扣钱交易错误与重试

针对txtype=500的Debit请求，也就是投注交易，若运营商的API伺服器因为non-deterministic错误而失败，该交易请求就会被UGS客户端API重试一次。

*非确定性的错误是交易没有清楚的成功或失败结果，包括“900 - System Error”的系统错误讯息，或网络超时联机中断。*

若重试仍然失败，UGS就会对游戏供应商针对此失败提出报告，然后将一个Cancel指令操作放置到排序(queue)结构中以便执行取消最后一笔的Debit请求。且UGS客户端API会做五次尝试取消交易。之后此取消动作会被设置进入排序中等待手动作业。

若错误是确定性的，例如“100 - Insufficient Funds”资金不足的讯息，那么扣钱交易就会立即失败，且不会被重试，且此错误会被立即回报给游戏供应商。

针对txtype数值不是500的Debit请求，错误处理和重试逻辑与下面所描的Credit请求是相同的。

### 加钱交易错误与重试

UGS客户端API会对加钱请求失败启动一次重试。若此加钱尝试失败，一个成功结果就会被返回给游戏供应商，包含加钱后预期的余额。

*这表示有些情况下，例如网络连接不稳定，可能会有游戏客户和运营商之间余额的差异。然而，一旦这些失败的指令操作(如Credit)结束且更新的余额被返回到游戏供应商之后，这个差异就会消失。*

然后加钱会对运营商作五次重试，之后就会被送进队列(queue)中等待手动作业。

## 操作失败摘要

	<b>Debit</b> 的投注指令(txttype=500)	<b>Credit</b> 的 Win、Tie、Cancel Place-Bet、FreeBet、Jackpot 指令。 <b>Debit</b> 的 Cancel Settlement 指令。
非确定性错误	<p>若Debit请求的失败原因无法确定 (non-deterministic)，UGS会立即重试一次，但若重试仍失败时，将会尝试取消这笔交易。</p> <p>方法：</p> <ol style="list-style-type: none"> <li>1. 重试加钱一次。</li> <li>2. 若失败，在UGS失败作业待执行排序中添加重试动作。</li> <li>3. 返回失败给游戏供应商</li> <li>4. 尝试取消五次。</li> </ol>	<p>若对运营商加钱请求失败，一个成功的值会返回给游戏供应商，且UGS会处理排序待行的作业，并对运营商进行重试作业。</p> <p>方法：</p> <ol style="list-style-type: none"> <li>1. 重试一次。</li> <li>2. 若失败，在UGS失败作业待执行排序中添加重试动作。</li> <li>3. 对游戏供应商返回成功结果，与估计余额(前次余额+加钱金额)。</li> <li>4. 对运营商作五次重试。</li> </ol>
确定性错误	失败 (将错误返回给游戏供应商)	

## 9. 直接启动游戏

UGS游戏启动程序包括两个组件：一个游戏列表API和游戏启动页。本节介绍运营商如何从UGS得到一个游戏大厅的游戏列表，和从游戏启动页中启动游戏。

先决条件：

- 在UGS后台系统中，创建一个属于运营商的品牌(权证持有者)。
- 在UGS后台系统中，为该品牌创建一个游戏大厅，并将游戏添加到这个大厅中。

运营商端请参阅以下操作步骤：

1. 运营商取得访问令牌(请参阅第5.1节)。
2. 运营商通过调用UGS游戏列表API取得游戏列表。请参阅第9.1节。
3. 运营商在他们的游戏大厅里展示从游戏列表API返回的游戏，与启动游戏的网址，如果玩家通过身份验证则可打开游戏(持有有效的玩家令牌)。请参阅 9.2节游戏启动页。

以下各节分别说明游戏列表API和游戏启动页。如果你想要看的上述步骤的完整示例，请跳至 9.3节。

## 9.1 游戏列表API

游戏列表API用于获取特定品牌的游戏列表。

### HTTP方法：

GET

### HTTP header:

Content-Type: application/json

Authorization: Bearer <[Access Token](#)>

### URL方法：

https://<TGPIIntegrationAPIServer>/**api/games**?lang={language-code}&platformtype={platform}&iconres={icon-resolution}

### 方法参数：

参数	说明	类型
<b>lang</b>	语言代码如：en-US, zh-CN。 字符串(ISO-639-1 and ISO3166-1)，请参阅附录D。	字符串 长度：5字符。
<b>platformtype</b>	玩家的平台。 请参阅 <a href="#">附录L-平台类型</a> 的所有值。	整数
<b>authtoken</b> (可选)	玩家身份验证令牌。需要玩家身份验证令牌以取得真人娱乐的投注限额列表。	字符串
<b>iconres</b> (可选)	游戏图标解析度。支持208x208, 263x153, 343x200, 520x520, 534x316, 910x367, 1242x580。	字符串

### 返回参数：

参数	说明	类型
<b>iconres</b>	游戏图标解析度。支持208x208, 263x153, 343x200, 520x520, 534x316, 910x367, 1242x580。	数组
<b>games</b>	包含下列字段的对象数组容器：	数组
<b>externalid</b>	游戏的外部id。	字符串 最大长度：30字符。
<b>code</b>	游戏代码，将用来启动游戏。	字符串 最大长度：30
<b>name</b>	在请求的'lang'参数中，明订游戏名称的语言文字。	字符串 最大长度：50
<b>description</b>	在请求的'lang'参数中，明订游戏说明的语言文字。	字符串 最大长度：2048
<b>providercode</b>	游戏供应商代码。	字符串 最大长度：30
<b>isactive</b>	True/False表示游戏的状态。若游戏或其供应商在维护中，此参数值为false；否则为true。	逻辑
<b>type</b>	游戏种类代码如：0=Slot, 1=Table Game。 所有游戏种类代码，请参阅 <a href="#">附录H</a> 。	整数
<b>iconurl</b>	游戏图标的URL。例如 //cdn-static.cdncache123.info/game/en-US/Game_TGP_Neptunes_Riches_343x200.jpg。 若iconres未传进来，将会返回一个虚值(null)。	字符串
<b>betlimits</b>	包含一个或多个投注限制的对象数组。	数组
投注限额数组的子对象。		
<b>id</b>	在直接启动游戏中提供的投注限制标识码。	字符串
<b>min</b>	投注限制的最小投注金额。	小数点 精确度：小数点两位
<b>max</b>	投注限制的最大投注金额。	小数点 精确度：小数点两位

示例：

### 请求(Request)

GET /api/games?lang=en-US&platformtype=0

#### Headers:

Content-Type: application/json

Authorization: Bearer pM8ZEKjbkn5kC7RqvdBn1gcZRfC8ccVxBbQmbIZYnVYeb4LirxfS9XARFvPJaoKjM

### 响应(Response)

请求成功的响应结果：

```
{
  "iconres": [
    "208x208",
    "263x153",
    "343x200",
    "520x520",
    "534x316",
    "910x367",
    "1242x580"
  ],
  "games": [
    {
      "externalid": "NeptunesRiches",
      "code": "Neptunes_Riches",
      "name": "Neptune's Riches",
      "description": "Follow the mystery bubbles and swim with the WILDS to unlock the infinite riches of Neptune!",
      "providercode": "TGP",
      "isactive": true,
      "type": 0,
      "iconurl":
        "//cdn-static.cdncache123.info/game/en-US/Game_TGP_Neptunes_Riches_343x200.jpg",
      "betlimits": null
    },
    {
      "externalid": "GoldenLotus",
      "code": "Golden_Lotus",
      "name": "Golden Lotus",

```

```

    "description": "Journey to the lake of golden petals and discover the wins, wilds and
bonuses hidden beneath!",
    "providercode": "TGP",
    "isactive": true,
    "type": 0,
    "iconurl":
"//cdn-static.cdncache123.info/game/en-US/Game_TGP_Golden_Lotus_343x200.jpg",
    "betlimits": null
},
{
    "externalid": "1",
    "code": "Baccarat_1",
    "name": "Baccarat 1",
    "description": "Both the Banker and the Player will receive at least 2, but no more
then 3 cards each time. If the first round ends in a draw, then the second round starts. The
one with points closest to 9 wins.",
    "providercode": "SB",
    "isactive": true,
    "type": 2,
    "iconurl": "//cdn-static.cdncache123.info/game/en-US/Game_SB_Baccarat_5_343x200.jpg",
    "betlimits": [
        {
            "id": "535",
            "min": 30,
            "max": 500
        },
        {
            "id": "561",
            "min": 30,
            "max": 1000
        },
        {
            "id": "587",
            "min": 100,
            "max": 10000
        }
    ]
},
...
]
}

```

请求失败的响应结果：

```
{
  "err": 10,
  "errdesc": "Token has expired"
}
```



## 9.2 游戏启动页

游戏启动页是让运营商启动游戏使用。请按照本节中所述的说明进行操作，运营商可以为每个游戏创建游戏启动网址，并使用该网址来启动游戏。

先决条件：

- 有效的玩家授权令牌，表示玩家持有有效的UGS玩家令牌，当玩家登陆运营商网站，可以打开游戏(请参阅6.3节)。
- 从游戏列表API取得所需的的游戏信息(请参阅9.1节：游戏列表API)。具体来说，要启动一个游戏，运营商需要提供游戏供应商代码、游戏代码、平台类型、玩家令牌，通过游戏启动页的网址来启动。

游戏启动页的网址：

`http://<TGP-Lobby-Domain>/gamelauncher?gpcode={game-provider-code}&gcode={game-code}&platform={platform-type}&token={player-token}`

其中<TGP-Lobby-Domain>可以是以下值：

- 暂存环境：staging.lobby188.info
- 生产环境：lobby188.info

参数：

名称	说明
<b>gpcode</b>	游戏供应商代码
<b>gcode</b>	游戏代码
<b>platform</b>	启动游戏的平台。 值： 0 = 桌面计算机 1 = 移动设备
<b>token</b>	玩家令牌
<b>betlimitid</b> (可选)	投注限额ID

## 示例：

[http://staging.lobby188.info/gamelauncher?gpcode=TGP&gcode=Dragons\\_Luck&platform=0&token=1TbGFBdezVduidEfzP98QZ78ZCY63bOZLxqaPE5x4u2MwPSwhfKOLE8QuCCpQQ7a4s1IIAJhDGo6MvOj57CoXEzI5Q78ruFFxkP7CwBzY1MJ83nrmFSmO5VkGIBxLjnbEo](http://staging.lobby188.info/gamelauncher?gpcode=TGP&gcode=Dragons_Luck&platform=0&token=1TbGFBdezVduidEfzP98QZ78ZCY63bOZLxqaPE5x4u2MwPSwhfKOLE8QuCCpQQ7a4s1IIAJhDGo6MvOj57CoXEzI5Q78ruFFxkP7CwBzY1MJ83nrmFSmO5VkGIBxLjnbEo)

[http://staging.lobby188.info/gamelauncher?gpcode=TGP&gcode=Dragons\\_Luck&platform=0&token=1TbGFBdezVduidEfzP98QZ78ZCY63bOZLxqaPE5x4u2MwPSwhfKOLE8QuCCpQQ7a4s1IIAJhDGo6MvOj57CoXEzI5Q78ruFFxkP7CwBzY1MJ83nrmFSmO5VkGIBxLjnbEo&betlimitid=535](http://staging.lobby188.info/gamelauncher?gpcode=TGP&gcode=Dragons_Luck&platform=0&token=1TbGFBdezVduidEfzP98QZ78ZCY63bOZLxqaPE5x4u2MwPSwhfKOLE8QuCCpQQ7a4s1IIAJhDGo6MvOj57CoXEzI5Q78ruFFxkP7CwBzY1MJ83nrmFSmO5VkGIBxLjnbEo&betlimitid=535)

若网址和参数正确，游戏启动页会将客户端的浏览器重定向到游戏的网址。

## 9.3 : 启动游戏的示例

这里有一个完整API请求/响应的示例，展示启动真人娱乐游戏的每一个步骤。包括：

1. 生成品牌访问令牌。
2. 生成玩家令牌(授权一名玩家)。
3. 使用游戏列表API取得游戏列表。
4. 从游戏启动页启动游戏。

下面的示例要求使用TGP作为品牌，且服务器为暂存环境。

### 9.3.1 生成品牌访问令牌

请求(Request)

POST <http://staging.tgpaccess.com/api/oauth/token>

Content-Type: application/json

```
{
  "client_id": "tgp",
  "client_secret": "pPTii7Jsg8Thzk1174hSD9h7D1b0VNdR2pXAjigV76Aq9egdLXTB9d",
  "grant_type": "client_credentials",
  "scope": "playerapi"
}
```

响应(Response)

```
{
  "access_token": "wgwPwspacMC0s80DUAUx8thZYvV0krkgKZCYeWZGdmi7pPEgxbMLzrWFzHuQPqy2",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "playerapi"
}
```

access\_token的值将在后续步骤中使用。

### 9.3.2 生成玩家令牌

请求(Request)

POST http://staging.tgpaccess.com/api/player/authorize

Authorization: Bearer wgwPwspacMC0s80DUAUx8thZYvV0krkgKZCYewZGdmi7pPEgxbbMLzrWFzHuQPqy2

Content-Type: application/json

```
{
  "ipaddress": "123.123.123.1",
  "username": "frank enstein",
  "userid": "frankenstein",
  "tag": null,
  "lang": "zh-CN",
  "cur": "RMB",
  "betlimitid": 1,
  "istestplayer": true
}
```

响应(Response)

```
{
  "authtoken":
  "TPvYxzEysPgHs81Edhj71TTQrnETLWvzZwcU9jkCK0wChhObZyftcPGHyNZCzbwFFp6t2Mc380pCbHi11FgrAEpLSFHuk
  Nd8BwubpEBCc5Pnrrmicf21iCJhzjmKIEMkEU",
  "isnew": true
}
```

### 9.3.3 使用游戏列表API取得游戏列表

请求(Request)

GET http://staging.tgpaccess.com/api/games? lang=en-US&platformtype=0

Authorization: Bearer wgwPwspacMC0s80DUAUx8thZYvV0krkgKZCYewZGdmi7pPEgxbbMLzrWFzHuQPqy2

Content-Type: application/json

响应(Response)

```
{
  "games": [
    {
      "code": "Baccarat_1",
      "name": "Baccarat 01",
      "description": "Both the Banker and the Player will receive at least 2....",
      "providercode": "SB",
      "isactive": true,
      "type": 2
    },
    {
      "code": "Multi_Bet_Baccarat_03",
      "name ": "Multi Bet Baccarat 03",
      "description": "A Baccarat game with an expanded betting area experience.",
      "providercode": "GD",
      "isactive": true,
      "type": 2
    },
    ... . (skipped)
  ]
}
```

### 9.3.4 从游戏启动页启动游戏

如前面所述，游戏启动页网址是：

`http://<TGP-Lobby-Domain>/gamelauncher?gpcode={game-provider-code}&gcode={game-code}&platform={platform-type}&token={player-token}`

因此，基于上述步骤得到的信息，启动Sunbet游戏"Baccarat\_1"的网址是：

`http://staging.lobby188.`

`info/gamelauncher?gpcode=SB&gcode=Baccarat_1&platform=0&token=TPvYxzEysPgHs81Edhj71TTQrnETLWvzZwcU9jkCK0wChhObZyftcPGHyNZCzbwFFp6t2Mc380pCbHi11FgrAEpLSFHukNd8BwubpEBCc5Pnrrmicf21iCJhzjmKIEMkEU`

# 附录

## 附录A - OAuth错误表

错误代码	说明	注记
invalid_request	无效的请求。	例如：在request中遗漏必要的属性。
unauthorized_client	未获授权客户端。	错误的客户标识码(client_id)或客户端密钥(client_secret)。
invalid_scope	无效的范围。	
server_error	服务器错误。	

## 附录B -支持HTTP的状态代码

状态代码	定义	注记
200	OK	成功执行request，且回传预期结果。
400	错误请求	遗漏Authorization Bearer header。
401	未获授权	例如：取得访问令牌失败或授权玩家失败。
404	找不到。	找不到资源。例如：错误的API端点或路径。
405	不允许的方法	使用错误的HTTP动词(GET, POST)或未支持的HTTP方法。
500	内部服务器错误	
503	暂停服务	

## 附录C - VIP级别

VIP级别标识码	说明
1	青铜-基本额度
2	白银-升级额度
3	黄金-高级额度
4	白金-贵宾额度
5	钻石-重要贵宾额度

## 附录D - 语言

下表显示TGP平台中定义的语言代码。

语言代码	说明
en-US	英文：默认
zh-CN	简体中文
zh-TW	繁体中文
ja-JP	日语
ko-KR	韩文(한 국 어 / 조 선 말 )
th-TH	泰文(ภาษาไทย)
vi-VN	越南文(ngôn ngữ tiếng Việt)
id-ID	印度尼西亚文(bahasa Indonesia)



## 附录E -货币

下表显示TGP平台中定义的货币代码。

货币代码	说明
RMB	人民币
EUR	欧元
USD	美元
CAD	加元
GBP	英镑
JPY	日元
THB	泰国铢
VND	越南盾
IDR	印尼盾
MYR	馬來西亞令吉
压缩币值(即VND/1000)可以根据要求创建。	

## 附录F -交易类型(txtype)

ID	说明	注记
500	投注(Place bet)	
510	赢钱(Win bet)	
520	输钱(Lose bet)	未使用。 0值表示交易未送出。
530	免费投注(Free bet)	
540	平手(Tie bet)	
550	调整交易	
560	取消交易	

## 附录G -奖金类型(bonustype)

ID	说明	注记
0	没有奖金	
1	奖金自旋	

## 附录H -游戏类型(gametype)

ID	说明	注记
0	老虎机(Slot)	
1	桌上游戏(Table Game)	
2	真人娱乐(Live Dealer)	
3	游戏大厅(Lobby)	包含许多不同的游戏。
4	立即赢钱(Instant Win)	如：5050、Link2Win、Scratch cards等游戏。
5	乐透(Lottery)	
6	迷你游戏-老虎机	
7	迷你游戏-即时战斗游戏	
8	迷你游戏-桌上游戏	

## 附录I -外部电子钱包集成的错误表

错误代码	说明	注记
12	无效或过期的访问令牌。	遇到此错误时，刷新以恢复访问令牌。
20	玩家被禁用。	
60	无效的金额。	对玩家的加钱或扣钱金额少于或等于0。
100	资金不足	玩家无足够资金作扣钱扣款。
300	无效参数。	例如：在request中遗漏必要的属性。
400	遗漏授权承载标头。	
500	错误的格式。	授权玩家时，无效的标签格式。
600	找不到玩家。	
900	系统错误。	

## 附录J -单一电子钱包集成的错误表

错误代码	说明	注记
10	无效或过期的令牌。	
19	无效令牌范围。	
20	屏蔽用户。	
30	无效的凭证。	
50	货币不符。	
100	资金不足，无法执行操作。	
300	无效参数: {参数名称或原因}	示例： 无效参数：用户不存在。
400	遗漏令牌。	
500	错误的格式。 {请进一步说明细节}	示例： 错误的格式：'2016-03-20 23:01.9999'
600	交易记录不存在。	
610	交易已被取消。	
800	确定性的操作失败。	
900	系统错误。	非确定性的操作失败。
903	超时(timeout)的组态设定。	

注记：针对错误描述，尽可能提供详细信息，这样才可较容易看出问题的症结所在以及如何修正问题。

## 附录K -游戏供应商列表

代码	游戏供应商
TGP	Red Tiger
SB	Sunbet
GB	Globalbet
GD	Gold Deluxe
LAX	Laxino

## 附录L -平台类型

代码	游戏供应商
0	桌面计算机
1	移动设备
4	迷你游戏

注意：没有代码为3的平台类型。

## 附录M -余额同步API

事件	定义
updateBalance {事件, 余额}	当玩家余额变动时, 发送事件。

### 示例：

现代浏览器以安全理由限制iframe和主窗口(main window)之间的直接通信。这就是为什么最好是使用此处所描述的postMessage API：

<https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage>

### 主窗口作业：

```
function receiveMessage(event){
    var data = event.data;

    if (data && data.event == 'updateBalance') {
        console.log('update balance: ', data.balance);
    }
}

window.addEventListener('message', receiveMessage, false);
```

## 修订记录

版本	日期	变更说明	作者
1.0	2015-02-02	初稿。	David Parchomchuk
1.1	2015-03-18	修正单向通讯。	David Parchomchuk
1.2	2015-03-19	更新玩家授权流程。	David Parchomchuk
1.3	2015-07-12	添加重试(retry)逻辑描述。	David Parchomchuk
1.4	2015-08-14	添加数据报告(data reporting)方法(method)。	David Parchomchuk
1.5	2015-10-21	在7.3节添加余额加总(balance total)方法。	Michael Tsai
1.6	2015-11-02	在7.6.4节添加历史/投注(history/bets)。 针对History API添加“includetestplayers”参数。	Michael Tsai
1.6.1	2015-12-11	自此版本开始，标签参数api/player/authorize若是从客户端传来，就必须是有效的JSON字符串。 针对一些API的成功回传，添加err和errdesc属性。 对样本调用(sample calls)格式作小幅变更以利可读性。	Michael Tsai
1.6.2	2015-12-28	添加用户名(username)到api/players/balance/list的结果(每一个AG请求(request))。 变更enddate为下列History API的专属结局：game、bettransaction、transfers、bets。	Michael Tsai
1.6.3	2016-01-20	在6.3节添加平台类型(platformtype)。	Pete Chen
1.6.4	2016-01-29	更新附录A中的错误代码。	Luke Hu
1.6.5	2016-02-05	修订并增加附录A及附录B的错误状态代码及HTTP状态代码。	Pete Chen
1.6.6	2016-03-04	针对history/bets参数StartDatefor的错误除错。	Michael Tsai
1.6.7	2016-03-30	添加history/bets参数issettled。 针对玩家授权和取消授权添加样本调用。 改进一些措词。	Michael Tsai
1.6.8	2016-04-12	- 在第4节添加一般性错误处理的说明。 - 从每一个API方法的描述移除重复的err和errdesc。 - 小修改(TOC，改变措词)。	Michael Tsai
1.6.9	2016-06-08	- 在Bet History方法针对'startdate'和'enddate'参数添加一些注记。 - 添加货币和语言代码附录的说明。 - 将KRD和HKD从8.5节的附录E -货币移除。	Michael Tsai
1.6.10	2016-06-20	- 强调'istestplayer'参数的重要性，确保读者了解此一特性。 - 变更“UGS”一词为“TGP”。部份描述内容目前尚未更动。	Michael Tsai

1.7.0	2016-07-22	<ul style="list-style-type: none"> <li>- 在"6.2节-游戏大厅端点"添加说明。</li> <li>- 在api/history/bets的返回资料中添加'playertype'。</li> </ul>	Michael Tsai
1.7.1	2016-08-03	<ul style="list-style-type: none"> <li>-更新繁体及简体中文翻译版本。</li> <li>-完成api/history/bets返回资料中的'playertype'。</li> </ul>	Andy So
1.7.2	2016-08-07	<ul style="list-style-type: none"> <li>-添加Single Wallet API方法。</li> <li>-添加文件说明，阐述外部电子钱包(External wallet)和单一电子钱包(Single wallet)集成的区别。</li> </ul>	David Parchomchuk
1.7.3	2016-09-02	<ul style="list-style-type: none"> <li>-添加遗漏的API URL: /wallet/token。</li> <li>-针对wallet/debit和wallet/credit方法的样本请求(sample request)，添加更多的参数。</li> </ul>	Michael Tsai
1.7.4	2016-09-08	<p>变更Isclosinground值，以true/false取代1/0。            添加响应失败参数和示例。            针对External Wallet的集成，将Table of Errors分开成Table of OAuth Errors和Table of Errors。            针对Single Wallet的集成，添加Table of Errors。</p>	Pete Chen
1.7.5	2016-09-10	<ul style="list-style-type: none"> <li>-修正样本请求中的gametype参数：“Slot”变更为0。</li> <li>-将'desc'字段从Proxy Wallet API移除。</li> <li>-针对Proxy Wallet API /wallet/cancel方法，将参数名称'ptxid'修正为'reftxid'。</li> </ul>	Michael Tsai
1.7.6	2016-10-03	<ul style="list-style-type: none"> <li>-添加'platformtype'参数到Proxy Wallet API的加钱与扣钱方法中。</li> </ul>	Michael Tsai
1.8.0	2016-10-14	<ul style="list-style-type: none"> <li>-添加第9节游戏启动器。</li> <li>-添加'ugstxid'到api/history/bettransaction的结果和样本响应。</li> <li>-添加'ugsroundid'到api/history/game的结果和样本响应。</li> </ul>	Michael Tsai
1.9.0	2016-10-26	添加9.3节启动游戏的示例。	Michael Tsai
1.10.0	2016-11-08	添加7.6.5节来自游戏供应商的游戏历史。	Alex Wei
1.10.1	2016-11-10	<p>为求一致性，改变单一电子钱包API交易顺序图的样式。            小修改gametype值的描述：1=LiveDealer。</p>	Michael Tsai
1.10.2	2016-11-21	<ul style="list-style-type: none"> <li>- 添加'gameprovidercode'到History API方法与其样本中，包括：history/[bets/game/bettransaction]。</li> <li>- 添加<a href="#">附录L-平台类型</a>和迷你游戏。此外，还修改其他地方'platform'有关的参数。</li> <li>- 添加遗漏的'authtoken'字段到Proxy Wallet方法。</li> <li>- 为单一电子钱包API的'transactions'节点添加更多的描述。</li> <li>- 改善'users'节点wallet/balance方法的描述。</li> </ul>	Michael Tsai
1.10.3	2016-11-24	<p>添加注释至wallet/debit方法中的txtype：若txtype=500，所有的交易应该是全部失败或全部成功。            更正程式码示例中的输入错误。</p>	Michael Tsai Pete Chen
1.11.0	2016-11-28	<ul style="list-style-type: none"> <li>- 针对Single Wallet API的Credit和Debit方法，将'platformtype'从顶层节点移到'transactions'节点。</li> <li>- 将“Game Launch Page”用词改成“Game Launcher”，并改以“Direct Game Launch”作为章节名称。</li> </ul>	Michael Tsai
1.12.0	2016-12-01	<ul style="list-style-type: none"> <li>- 变更wallet/balance参数：现在运营商可返回本地化的钱包名称和描述，且当'walletcode'是空值时，可返回所有的钱包。</li> </ul>	Michael Tsai
1.13.0	2016-12-02	<ul style="list-style-type: none"> <li>- 添加Transaction Grouping至Single Wallet章节中。</li> <li>- 将中文版修订记录移到文档末尾处。(Michael)</li> </ul>	David Parchomchuk
1.13.1	2016-12-05	<ul style="list-style-type: none"> <li>- 添加'transactional'至Proxy Wallet API的Debit方法中。</li> </ul>	Michael Tsai



		- 添加'desc'至Proxy Wallet API的Credit和Deibt方法中。	
1.14	2016-12-06	- 添加'iconres'至游戏列表的请求查询字符串中。 - 添加'iconurl'至游戏列表的JSON回应中。	Pete Chen
1.14.1	2016-12-08	将Adjust Transaction从附件- F中移除。	Pete Chen
1.15.0	2016-12-09	- 将'refptxid'参数添加至Debit和Credit方法中。 - 小修改附录- J的内容与历史/投注的描述。	Michael Tsai
1.15.1	2016-12-14	- 改进Single Wallet API的ggr、turnover、unsettledbets描述。 - 改进第8.5节- Cancel的描述(以避免Cancel操作和Cancel交易的混淆)。 - 更新第8.8节-错误和重试处理。 - 添加Debit和Credit方法参数的最大长度。	Michael Tsai
1.15.2	2016-12-15	- 添加第8.8节-返回错误响应。 - 添加Single Wallet API的Get Balance方法的失败示例。 - 使字体样式更趋一致。	Michael Tsai
1.15.3	2016-12-22	- 将附录- F中不正确的陈述删除：txtype 520未使用。理由：对Single Wallet API而言，即使Credit操作的金额是零，LoseBet仍会被送出。 - 添加更多Single Wallet API的Debit和Credit方法的响应示例。	Michael Tsai
1.15.4	2017-01-11	-添加'betlimits'对象数组至Direct Game Launch的Game List API中。 -添加'externalid'至Direct Game Launch的Game List API中。	Owen Chen
1.15.5	2017-02-11	-将FlyCow从游戏供应商资料表中移除。 -将FlyCow从游戏供应商的游戏历史中移除。	Pete Chen
1.15.6	2017-02-15	-修复和审查每一个调用范例的url。	Maze Chang
1.15.7	2017-02-23	- 更新域名<IntegrationURL>为<TGPIIntegrationAPIServer>	Luke Hu
1.15.8	2017-02-25	- 修正API范例上的时间戳格式	Pete Chen
1.15.9	2017-03-07	- 更新ggr、turnover和unsettledbets的精确度	Jeff He
1.15.10	2017-03-24	- 更新url参数日期格式为url编码后格式	Luke Hu
1.15.11	2017-03-24	- 移除多余roundstatus状态New	Owen Chen
1.15.12	2017-03-27	- 在附录E中新增馬來西亞令吉	Pete Chen
1.15.13	2017-03-28	- 更新第8.9节单一电子钱包重试处理	David Parchomchuk
1.15.14	2017-03-30	- 更新第7.6.3节riskamt的定义 - 更新第7.6.4节riskamt、winloss和timestamp的定义 - 更新第8.3节timestamp和senton的定义 - 更新第8.4节timestamp和senton的定义 - 更新第8.5节timestamp的定义	Pete Chen