

PG SOFT

# 对接文档

外部应用编程接口 (External API) v1.8



## 内容

更新记录 .....	4
1. 概述 .....	11
2. API 对接模式 .....	12
<b>2.1 免费游戏</b> [最新版本: v1] .....	13
2.1.1 获取免费游戏 .....	13
2.1.2 获取已转换为 <b>红利游戏/现金</b> 的免费游戏列表 .....	18
<b>2.1.3 获取已转换为红利游戏并再转为现金的免费游戏列表</b> .....	20
2.1.4 获取免费游戏玩家 .....	23
2.1.5 获取免费游戏的未注册玩家 .....	26
2.1.6 获取免费游戏的所有玩家详情 .....	29
2.1.7 获取免费游戏的单个玩家详情: .....	33
2.1.8 创建免费游戏 .....	37
2.1.9 按投注金额创建免费游戏 .....	40
2.1.10 将玩家转移到固定免费游戏中 .....	42
2.1.11 将玩家转移到弹性免费游戏中 .....	44
2.1.12 取消免费游戏 .....	47
<b>2.1.13 取消玩家的免费游戏</b> .....	48
2.1.14 取消未注册玩家的免费游戏 .....	49
2.1.15 取消玩家的所有免费游戏 .....	50
<b>2.2 红利</b> [最新版本: v1] .....	51
<b>2.2.1 获取红利游戏列表</b> .....	51
2.2.2 获取已转换为现金的 <b>红利游戏</b> 列表 .....	55
2.2.3 获取 <b>红利游戏</b> 玩家 .....	57
2.2.4 获取未注册的 <b>红利游戏</b> 玩家 .....	60
2.2.5 获取红利游戏的所有玩家详情 .....	63
2.2.6 获取红利游戏的单个玩家详情 .....	66
2.2.7 创建 <b>红利游戏</b> .....	69

2.2.8	将玩家转移到固定红利游戏中 .....	71
2.2.9	将玩家转移到弹性红利游戏中 .....	73
2.2.10	取消红利游戏 .....	76
2.2.11	取消已注册玩家的红利游戏 .....	77
2.2.12	取消未注册玩家的红利游戏 .....	78
2.2.13	取消玩家的所有红利游戏 .....	79
<b>2.3</b>	<b>锦标赛 [最新版本: v2] .....</b>	<b>80</b>
2.3.1	获取锦标赛 .....	80
2.3.2	获取锦标赛玩家 .....	84
2.3.3	获取锦标赛的最高排名 .....	87
2.3.4	注册锦标赛玩家 .....	90
2.3.5	获取锦标赛的现金奖励列表 .....	92
<b>2.4</b>	<b>投注历史记录 [最新版本: v3] .....</b>	<b>95</b>
2.4.1	获取历史记录 .....	96
2.4.2	获取特定时间内的历史记录 .....	100
2.4.3	获取单一玩家的历史记录 .....	104
2.4.4	获取玩家的每日汇总 .....	107
2.4.5	获取玩家在特定时间内的每日汇总 .....	110
2.4.6	获取投注汇总 .....	113
2.4.7	获取每小时投注汇总 .....	116
2.4.8	获取玩家的投注总和 .....	118
2.4.9	获取玩家未完成的投注 .....	121
2.4.10	获取投注详情 .....	124
2.4.11	获取失败投注 .....	127
2.4.12	待处理投注 .....	129
2.4.12.1.	获取待处理投注 .....	131
2.4.12.2.	手动解决待处理投注 .....	133
2.4.13	手动重置待处理游戏状态 .....	135
<b>2.5</b>	<b>游戏 [最新版本: v2] .....</b>	<b>136</b>

2.5.1	获取游戏列表.....	136
2.5.2	获取游戏的合法投注金额.....	139
2.5.3	更改游戏发布状态 .....	141
<b>2.6</b>	<b>累积奖池 [最新版本: v1] .....</b>	<b>142</b>
2.6.1	获取累积奖池列表 .....	142
<b>2.7</b>	<b>玩家 [最新版本: v3] .....</b>	<b>144</b>
2.7.1	获取在线玩家人数 .....	144
2.7.2	获取玩家的线上状态 .....	146
<b>2.7.3</b>	<b>获取在线玩家列表 .....</b>	<b>147</b>
<b>2.7.4</b>	<b>创建玩家 .....</b>	<b>149</b>
<b>2.7.5</b>	<b>踢出玩家 .....</b>	<b>151</b>
<b>2.7.6</b>	<b>禁止玩家 .....</b>	<b>152</b>
<b>2.7.7</b>	<b>恢复玩家 .....</b>	<b>153</b>
<b>2.7.8</b>	<b>检查玩家状态 .....</b>	<b>154</b>
3.	附录 .....	155
<b>3.1</b>	<b>平台 .....</b>	<b>155</b>

## 更新记录

版本	日期	说明
1.0	2019-04-03	初稿
1.1	2019-04-12	添加新内容 <ul style="list-style-type: none"> <li>2.1.5 获取免费游戏的未注册玩家</li> <li>2.1.12 取消已注册玩家的免费游戏</li> <li>2.2.4 取消未注册的红利玩家</li> <li>2.2.11 取消已注册玩家的红利</li> <li>2.4.6 取消投注汇总</li> </ul> 修订概述 修订 API 请求参数 <ul style="list-style-type: none"> <li>2.1.6 创建免费游戏</li> <li>2.1.11 取消为主场玩家的免费游戏</li> <li>2.2.5 创建红利</li> <li>2.2.10 取消未注册玩家的红利</li> </ul> 修订 API 说明 <ul style="list-style-type: none"> <li>2 API 对接模式</li> <li>2.1.2 获取已转换为红利的免费游戏列表</li> <li>2.1.3 获取已转换为现金的免费游戏列表</li> <li>2.1.10 取消已注册玩家的免费游戏</li> <li>2.1.11 取消未注册玩家的免费游戏</li> <li>2.2.2 获取已转换为现金的红利列表</li> <li>2.2.9 取消已注册玩家的红利</li> <li>2.2.10 取消未注册玩家的红利</li> </ul>
1.2	2019-04-15	为所有投注历史的 API 添加了注意 (2.4)
1.3	2019-04-17	添加新内容 <ul style="list-style-type: none"> <li>2.1.6 获取免费游戏的所有玩家</li> <li>2.1.7 为玩家获取免费游戏</li> <li>2.2.5 获取所有红利玩家</li> <li>2.2.6 获取玩家的红利</li> </ul> 修订 API 说明 & 请求说明 <ul style="list-style-type: none"> <li>2.1.4 获取免费游戏玩家</li> <li>2.1.5 获取免费游戏的未注册玩家</li> </ul>

		<ul style="list-style-type: none"> <li>2.2.3 获取红利玩家</li> <li>2.1.5 获取未注册的红利玩家</li> <li>2.4.6 获取投注汇总</li> <li>2.5.1 获取游戏列表</li> </ul> 修订 API 请求 <ul style="list-style-type: none"> <li>2.1.5 获取免费游戏的未注册玩家</li> <li>2.1.7 创建免费游戏</li> <li>2.2.4 获取未注册的红利玩家</li> <li>2.2.6 创建红利</li> </ul> 修订 API 响应 <ul style="list-style-type: none"> <li>2.1.6 获取免费游戏的所有玩家</li> <li>2.1.7 创建免费游戏</li> <li>2.2.5 获取所有红利玩家</li> <li>2.2.6 创建红利</li> <li>2.4.6 获取投注汇总</li> <li>2.5.2 获取游戏列表</li> </ul>
1.4	2019-07-12	添加新内容 <ul style="list-style-type: none"> <li>2.1.9 按投注金额创建免费游戏</li> <li>2.3.1 获取锦标赛</li> <li>2.3.2 获取锦标赛玩家</li> <li>2.5.2 获取游戏的合法投注金额</li> </ul> 修订 API 说明 <ul style="list-style-type: none"> <li>2.5.1 获取游戏列表 (v2)</li> </ul> 修订 API 请求 <ul style="list-style-type: none"> <li>2.1.8 创建免费游戏</li> </ul> 修订 API 响应 <ul style="list-style-type: none"> <li>2.1.1 获取免费游戏</li> </ul>
1.5	2019-08-22	修订 API 说明 <ul style="list-style-type: none"> <li>2.1.1 获取免费游戏</li> <li>2.1.2 获取已转换为红利/现金的免费游戏列表</li> <li>2.1.3 获取已转换为现金的免费游戏红利列表</li> <li>2.1.4 获取免费游戏玩家</li> <li>2.1.6 获取免费游戏的所有玩家</li> <li>2.2.1 获取红利</li> </ul>

		<ul style="list-style-type: none"> <li>2.2.2 获取已转换为现金的红利列表</li> <li>2.2.3 获取红利玩家</li> <li>2.2.5 获取所有红利玩家</li> <li>2.3.1 获取锦标赛</li> <li>2.3.2 获取锦标赛玩家</li> <li>2.4.5 获取玩家在特定时间内的每日汇总</li> </ul> <p>修订 API 请求</p> <ul style="list-style-type: none"> <li>2.1.7 为玩家获取免费游戏</li> <li>2.2.9 将玩家转移到弹性红利中</li> <li>2.4.1 获取历史记录</li> </ul> <p>修订 API 响应</p> <ul style="list-style-type: none"> <li>2.4.1 获取历史记录</li> <li>2.4.2 获取特定时间内的历史记录</li> <li>2.4.3 获取单一玩家的历史记录</li> <li>2.4.6 获取投注汇总</li> </ul> <p>添加新内容</p> <ul style="list-style-type: none"> <li>2.4.7 获取玩家的投注总和</li> </ul>
1.6	2019-11-13	<p>添加新内容</p> <ul style="list-style-type: none"> <li>2.4.7 获取每小时投注汇总</li> </ul> <p>修订 API 版本</p> <ul style="list-style-type: none"> <li>2.4 投注历史记录</li> </ul> <p>修订 API 说明</p> <ul style="list-style-type: none"> <li>2.4 投注历史记录</li> </ul> <p>修订 API 说明</p> <ul style="list-style-type: none"> <li>2.4.1 投注历史记录</li> </ul> <p>修订 API 响应</p> <ul style="list-style-type: none"> <li>2.4.1 获取历史记录</li> </ul> <p>修订 API 响应</p> <ul style="list-style-type: none"> <li>2.4.2 获取特定时间的历史记录</li> </ul> <p>修订 API 响应</p> <ul style="list-style-type: none"> <li>2.4.3 获取单个玩家的历史记录</li> </ul>
1.7	2020-02-10	<p>添加新内容</p> <ul style="list-style-type: none"> <li>2.5.3 更改游戏发布状态</li> <li>2.7.1 获取在线玩家人数</li> </ul>

		<ul style="list-style-type: none"> <li>2.7.2 获取玩家线上状态</li> </ul> 修订 API 响应 <ul style="list-style-type: none"> <li>2.4.1 获取历史记录</li> <li>2.4.2 获取特定时间的历史记录</li> <li>2.4.3 获取单个玩家的历史记录</li> <li>2.4.4 获取玩家的每日汇总</li> <li>2.4.5 获取玩家在特定时间内的每日汇总</li> <li>2.4.6 获取投注汇总</li> <li>2.4.7 获取每小时投注汇总</li> <li>2.4.8 获取玩家的投注总和</li> <li>2.6.1 获取累积奖池列表</li> </ul> 修订 API 响应参数说明 <ul style="list-style-type: none"> <li>2.1.6 获取免费游戏的所有玩家详情</li> <li>2.1.7 获取免费游戏的单个玩家详情</li> <li>2.2.5 获取红利游戏的所有玩家详情</li> <li>2.2.6 获取红利游戏的单个玩家详情</li> </ul>
1.71	2020-02-27	修订 API 请求 <ul style="list-style-type: none"> <li>2.4.5 获取玩家在特定时间内的每日汇总</li> </ul>
1.8	2020-12-11	添加新的 API 版本 <ul style="list-style-type: none"> <li>2.7 玩家</li> </ul> 添加新内容 <ul style="list-style-type: none"> <li>2.3.5 获取锦标赛的现金奖励列表</li> <li>2.4.9 获取玩家未完成的投注</li> <li>2.4.10 获取投注详情</li> <li>2.4.11 获取失败投注</li> <li>2.4.12 待处理投注               <ul style="list-style-type: none"> <li>2.4.12.1. 获取待处理投注</li> <li>2.4.12.2. 手动解决待处理投注</li> </ul> </li> <li>2.4.13 手动重置待处理游戏状态</li> <li>2.7.3 获取在线玩家列表</li> <li>2.7.4 创建玩家</li> </ul>



- 2.7.5 踢出玩家
- 2.7.6 冻结玩家
- 2.7.7 恢复玩家
- 2.7.8 检查玩家状态

#### 修订 API 标题或说明

- 2.1.3 获取已转换为红利游戏并再转为现金的免费游戏列表
- 2.1.10 将玩家转移到固定免费游戏中
- 2.1.13 取消玩家的免费游戏
- 2.2.6 获取红利游戏的单个玩家详情
- 2.2.7 创建红利游戏
- 2.4 投注历史记录
- 2.4.1 获取历史记录
- 2.4.2 获取特定时间内的历史记录
- 2.4.3 获取单一玩家的历史记录
- 2.4.4 获取玩家的每日汇总
- 2.4.5 获取玩家在特定时间内的每日汇总
- 2.4.6 获取投注汇总
- 2.4.7 获取每小时投注汇总
- 2.4.8 获取玩家的投注总和
- 2.4.9 获取玩家未完成的投注
- 2.4.10 获取投注详情
- 2.4.11 获取失败投注
- 2.4.12 待处理投注
- 2.6.1 获取累积奖池列表
- 2.7.2 获取玩家的线上状态

#### 修订 API 请求或响应内容

- 2.1.1 获取免费游戏
- 2.1.4 获取免费游戏玩家

- 2.1.5 获取免费游戏的未注册玩家
- 2.1.6 获取免费游戏的所有玩家详情
- 2.1.7 获取免费游戏的单个玩家详情
- 2.1.8 创建免费游戏
- 2.1.9 按投注金额创建免费游戏
- 2.1.10 将玩家转移到固定免费游戏中
- 2.1.11 将玩家转移到弹性免费游戏中
- 2.2.1 获取红利游戏
- 2.2.2 获取已转换为现金的红利游戏列表
- 2.2.3 获取红利游戏玩家
- 2.2.4 获取未注册的红利游戏玩家
- 2.2.5 获取红利游戏的所有玩家详情
- 2.2.6 获取红利游戏的单个玩家详情
- 2.2.9 将玩家转移到弹性红利游戏中
- 2.2.13 取消玩家的所有红利游戏
- 2.3.1 获取锦标赛
- 2.3.2 获取锦标赛玩家
- 2.3.4 注册锦标赛玩家
- 2.3.5 获取锦标赛的现金奖励列表
- 2.4.1 获取历史记录
- 2.4.2 获取特定时间内的历史记录
- 2.4.3 获取单一玩家的历史记录
- 2.4.4 获取玩家的每日汇总
- 2.4.5 获取玩家在特定时间内的每日汇总
- 2.4.6 获取投注汇总
- 2.4.7 获取每小时投注汇总
- 2.4.8 获取玩家的投注总和
- 2.4.9 获取玩家未完成的投注
- 2.4.10 获取投注详情
- 2.4.11 获取失败投注
- 2.4.12 待处理投注

2.6.1 获取累积奖池列表

2.7.2 获取玩家的线上状态

## 1. 概述

外部 API 让您得以透过编程方式使用 PGSoft 工具和服务。您可以通过这些 API 检索信息，并创建、修改或删除 PG 所提供的各种服务，例如免费游戏、**红利游戏**和锦标赛等。

PGSoft 的 API 皆按服务划分。每种服务在本文档中都有个别的章节。本文档所包含的所有 API 都是可选的，运营商可依据需求选择 API 对接。

## 2. API 对接模式

### 请求

外部 API 使用了 HTTP 表单请求方式及 RESTful 端点结构。请使用以下内容类型定义请求格式：

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

### 返回

API 的返回结果为 JSON 格式。系统将返回 HTTP 状态码 200 以表示成功与失败的请求。h

### 所有 API 方式

**Content-Type:** application/json

参数：

参数名称	数据类型	说明
<b>data</b> <sup>1</sup>	JSON Object	API 对接模式的响应信息
<b>error</b> <sup>2</sup>	JSON Object	发生错误或异常时的错误信息

例子：

成功的返回结果

```
{
  "data": {
    [API method response. JSON object format may vary depending on API methods]
  },
  "error": null,
}
```

失败/错误的返回结果

```
{
  "data": null,
  "error": {
    "code": "[Error code]",
    "message": "[Error message]"
  }
}
```

<sup>1</sup> 此栏目将在错误响应中显示空值。

<sup>2</sup> 此栏目将在成功应用编程接口响应中显示空值。请参阅有关[错误响应](#)的章节以了解更多关于错误响应的格式。

## 2.1 免费游戏 [最新版本: v1]

### 2.1.1 获取免费游戏

获取已创建的免费游戏列表:

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGames

HTTP 方法 : POST

参数

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
from_time	Long	否	免费游戏的开始时间范围 (以毫秒为单位的 Unix 时间戳)
to_time	Long	否	免费游戏的结束时间范围 (以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	免费游戏的创建时间范围 (以毫秒为单位的 Unix 时间戳)
to_created_time	Long	否	免费游戏的创建时间范围 (以毫秒为单位的 Unix 时间戳)
free_game_id	Integer	否	免费游戏的唯一标识符
currency	String	否	免费游戏的货币
free_game_name	String	否	免费游戏的名称
free_game_transfer_type	Integer	否	免费游戏的转换类型 0: 固定 3: 弹性
conversion_type	Char	否	完成免费游戏后, 红利的转换类型 C: 现金

			B: 红利
status	Integer	否	免费游戏的状态 0: 无效 1: 活跃 2: 已过期

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&from_time=1346592723000&to_time=1546592723000&from_created_time=1346592723000&to_created_time=1546592723000&free_game_id=12345&currency=CNY&free_game_name=freegame2019&free_game_transfer_type=1&conversion_type=C&status=1
```

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏列表

JSON 数组

参数名称	数据类型	说明
operatorToken	String	运营商的唯一标识符
transaction_id	String	交易的唯一标识符
freeGameId	Integer	免费游戏的唯一标识符
freeGameName	String	免费游戏的名称
isEvent	Boolean	免费游戏类型的指标 True: 弹性 False: 固定
gameIds	Integer Array	游戏的唯一标识符
totalGame	Integer	每位玩家将获得的免费游戏总数
currencyCode	String	免费游戏的货币
coinSize	Decimal	免费游戏的硬币大小

multiplier	Integer	免费游戏的投注倍数
minimumConversionAmount	Decimal	免费游戏的最低现金转换金额
maximumConversionAmount	Decimal	免费游戏的最高现金转换金额
conversionType	Char	完成免费游戏后，红利的转换类型 C: 现金 B: 红利
bonusRatio	Decimal	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后，需要完成的免费游戏投注额倍数)
bonusRatioAmount	Decimal	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后，需要完成的总金额)
bonusMinimumConversionAmount	Decimal	红利游戏的最低现金转换金额
bonusMaximumConversionAmount	Decimal	红利游戏的最高现金转换金额
bonusType	Integer	红利游戏的转换类型： 0: 红利 1: 现金 2: 红利 & 现金
status	Integer	免费游戏的状态 0: 无效 (已取消) 1: 活跃 2: 已过期
freeGameCountGiven	Integer	给予免费游戏的总数
convertedToCashAmount	Integer	已转换成现金的红利总额
convertedToBonusAmount	Integer	已转换成红利的红利总额
numberOfPlayers	Integer	免费游戏中的玩家总数
expiredDate	Long	免费游戏的截止日期 (以毫秒为单位的 Unix 时间戳)



createdDate	Long	免费游戏的创建日期（以毫秒为单位的 Unix 时间戳）
createdBy	String	创建免费游戏的 API 或 BackOffice 用户
updatedBy	String	更新免费游戏的 API 或 BackOffice 用户
isSupressDiscard	Boolean	允许玩家取消优惠的权限 True:不允许玩家取消优惠 False:允许玩家取消优惠

例子：

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "operatorToken": "abcd",
        "freeGameId": 1,
        "freeGameName": "free game1",
        "transactionId": "12345Test",
        "isEvent": false,
        "gameIds": [1],
        "totalGame": 20,
        "currencyCode": "USD",
        "coinSize": 0.05,
        "multiplier": 2,
        "minimumConversionAmount": 1000.00,
        "maximumConversionAmount": 1000.00;
        "conversionType": "B",
        "bonusRatio": 2,
        "bonusRatioAmount": 0,
        "bonusMinimumConversionAmount": 1000.00,
        "bonusMaximumConversionAmount": 1500.00;
        "bonusType": 0,
        "status": 2,
        "freeGameCountGiven": 0,
        "numberOfPlayers": 0,
        "expiredDate": 1499764521000,
        "createdDate": 1498036782000,
        "createdBy": "player1",
        "updatedBy": "player1",
        "isSupressDiscard": true
      },
      {
        "operatorToken": "abcd",
        "freeGameId": 2,
        "freeGameName": "free game2",
        "transactionId": "12345Test2",
```

```
        "isEvent": false,  
        "gameIds": [1],  
        "totalGame": 20,  
        "currencyCode": "USD",  
        "coinSize": 0.05,  
        "multiplier": 2,  
        "conversionType": "C",  
        "bonusRatio": 0,  
        "bonusRatioAmount": 0,  
        "status": 1,  
        "freeGameCountGiven": 0,  
        "numberOfPlayers": 0,  
        "expiredDate": 1499732101000,  
        "createdDate": 1498114200000,  
        "createdBy": "player1",  
        "updatedBy": "player1",  
        "isSupressDiscard": true  
    }  
]  
,  
"error": null  
}
```

## 2.1.2 获取已转换为红利游戏/现金的免费游戏列表

获取已转换为红利游戏/现金的免费游戏列表：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameConvertedWallet

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
free_game_id	Integer	是	免费游戏的唯一标识符
player_name	String	否	玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_transaction_time	Long	否	免费游戏玩家交易的时间范围 (以毫秒为单位的 Unix 时间戳)
to_transaction_time	Long	否	免费游戏玩家交易的时间范围 (以毫秒为单位的 Unix 时间戳)

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&free\_game\_id=12345&player\_name=player1&from\_transaction\_time=1346592723000&to\_transaction\_time=1346592723000

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalConvertedAmount	Decimal	已通过免费游戏转换成现金的总金额
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	免费游戏玩家钱包的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
gameId	Integer	游戏的唯一标识符
transactionAmount	Decimal	已转换的玩家总数
transactionDateTime	Long	转换时间

例子:

```
{
  "data": {
    "totalConvertedAmount": 10,
    "totalCount": 1,
    "totalPage": 1,
    "result": [
      {
        "balanceId": 13787,
        "playerName": "qa18",
        "currencyCode": "CNY",
        "gameId": 8,
        "transactionAmount": 10,
        "transactionDateTime": 1547449915000
      },
      {
        "balanceId": 13788,
        "playerName": "qa18",
        "currencyCode": "CNY",
        "gameId": 8,
        "transactionAmount": 10,
        "transactionDateTime": 1547449915100
      }
    ]
  },
  "error": null
}
```

### 2.1.3 获取已转换为红利游戏并再转为现金的免费游戏列表

获取已从免费游戏转换为红利游戏，并再从红利游戏转换为现金的游戏列表：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameBonusConvertedWallet

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
free_game_id	Integer	是	免费游戏的唯一标识符
player_name	String	否	玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_transaction_time	Long	否	免费游戏玩家交易的时间范围 (以毫秒为单位的 Unix 时间戳)
to_transaction_time	Long	否	免费游戏玩家交易的时间范围 (以毫秒为单位的 Unix 时间戳)
status	Integer	否	<b>红利游戏状态</b> 0: 无效 1: 活跃 2: 已过期

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&free\_game\_id=12345&player\_name=player1&from\_transaction\_time=1346592723000&to\_transaction\_time=1346592723000&status=1

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalConvertedAmount	Decimal	已通过免费游戏转换为现金的总金额
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	免费游戏玩家钱包的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
gameId	Integer	游戏的唯一标识符
transactionAmount	Decimal	已转换的玩家总数
transactionDateTime	Long	转换时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": {
    "totalConvertedAmount": 10,
    "totalCount": 1,
    "totalPage": 1,
    "result": [
      {
        "balanceId": 13787,
        "playerName": "qa18",
        "currencyCode": "CNY",
        "gameId": 8,
        "transactionAmount": 10,
        "transactionDateTime": 1547449915000
      },
      [
        {
          "balanceId": 13788,
          "playerName": "qa18",
          "currencyCode": "CNY",
          "gameId": 8,
          "transactionAmount": 10,
          "transactionDateTime": 1547449915100
        }
      ]
    ],
    "error": null
  }
}
```

## 2.1.4 获取免费游戏玩家

获取免费游戏的所有已注册玩家：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGamePlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
free_game_id	Integer	是	免费游戏的唯一标识符
player_name	String	否	免费游戏玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_time	Long	否	免费游戏的开始时间范围
to_time	Long	否	(以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	玩家被注册到免费游戏的时间范围
to_created_time	Long	否	(以毫秒为单位的 Unix 时间戳)
free_game_name	String	否	免费游戏的名称
status	Integer	否	免费游戏钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_name=player1&from_time=1346592723000&to_time=1546592723000&from_created_time=1346592723000&to_created_time=1546592723000&free_game_id=12345&free_game_name=freegame2019&status=1
```



## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	免费游戏玩家钱包的唯一标识符
playerId	Integer	免费游戏玩家的唯一标识符
playerName	String	免费游戏玩家的唯一标识符
walletType	String	免费游戏玩家的钱包类型 G: 免费游戏 B: 红利
balanceAmount	Decimal	玩家的免费游戏余额
bonusRatioAmount	Decimal	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后, 需要完成的总金额)
bonusMaximumConversionAmount	Decimal	免费游戏红利的最高转换金额
freeGameMaximumConversionAmount	Decimal	免费游戏的最高转换金额
status	Integer	免费游戏钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
expiredDate	Long	免费游戏的截止日期 (以毫秒为单位的 Unix 时间戳)

createdBy	String	创建免费游戏的 API 或 BackOffice 用户
updatedBy	String	更新免费游戏的 API 或 BackOffice 用户
updatedAt	Long	玩家被注册或重新注册到免费游戏的日期 (以毫秒为单位的 Unix 时间戳)
createdAt	Long	玩家被注册到免费游戏的日期 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "balanceId": 111261,
        "playerId": 7443,
        "playerName": "player1",
        "walletType": "B",
        "balanceAmount": 874.4,
        "bonusRatioAmount": 874.4,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdBy": "operator",
        "updatedBy": "operator",
        "createdAt": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "freeGameCount": 6,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdBy": "operator",
        "updatedBy": "operator",
        "createdAt": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

## 2.1.5 获取免费游戏的未注册玩家

获取待注册的免费游戏玩家：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGamePendingPlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
free_game_id	Integer	是	免费游戏的唯一标识符
player_name	String	否	免费游戏未注册玩家的唯一标识符 <b>注：实行模糊匹配</b>
created_by	String	否	注册玩家的 API 或 BackOffice 用户
status	Integer	否	待定钱包的状态 0: 无效（已取消） 1: 待处理 2: 已转移

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&player\_name=player1&free\_game\_id=12345&created\_by=external&status=1

## 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏未注册玩家列表

JSON 数组

参数名称	数据类型	说明
pendingWalletId	Integer	免费游戏玩家待定钱包的唯一标识符
playerName	String	免费游戏未注册玩家的唯一标识符
status	Integer	待定钱包的状态 0: 无效（已取消） 1: 待处理 2: 已转移
updatedAt	Long	玩家被注册或重新注册到免费游戏的日期 (以毫秒为单位的 Unix 时间戳)
createdAt	Long	玩家被注册到免费游戏的日期 (以毫秒为单位的 Unix 时间戳)
createdBy	String	注册玩家的 API 或 BackOffice 用户
updatedAt	String	注册或重新注册玩家的 API 或 BackOffice 用户

例子:

```
{
  "data": {
    "totalCount": 2,
    "totalPage": 1,
    "result": [
      {
        "pendingWalletId": 140392,
        "playerName": "eddywang9999",
        "status": 1,
        "createdDate": 1554351552000,
        "updatedAt": 1554351552000,
        "createdBy": "External API",
        "updatedAt": "External API"
      },
      {
        "pendingWalletId": 140393,
        "playerName": "eddywang99999",
        "status": 1,
        "createdDate": 1554351682000,
        "updatedAt": 1554351682000,
        "createdBy": "External API",
        "updatedAt": "External API"
      }
    ]
  },
  "error": null
}
```

## 2.1.6 获取免费游戏的所有玩家详情

获取免费游戏的所有玩家的详细信息：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameAllPlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
free_game_id	Integer	是	免费游戏的唯一标识符
player_name	String	否	免费游戏未注册玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_time	Long	否	免费游戏的开始时间范围 (以毫秒为单位的 Unix 时间戳)
to_time	Long	否	免费游戏的结束时间范围 (以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	免费游戏的创建时间范围 (以毫秒为单位的 Unix 时间戳)
to_created_time	Long	否	免费游戏的创建时间范围 (以毫秒为单位的 Unix 时间戳)
free_game_name	String	否	免费游戏的名称
status	Integer	否	免费游戏钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&p
layer_name=player1&from_time=1346592723000&to_time=1546592723000&from_created_time
=1346592723000&to_created_time=1546592723000&free_game_id=12345&free_game_name=fre
e game2019&status=1
```

## 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏未注册玩家列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	免费游戏玩家钱包的唯一标识符
playerId	Integer	免费游戏玩家的唯一标识符
playerName	String	免费游戏玩家的唯一标识符
walletType	String	玩家的钱包类型 G: 免费游戏 B: 红利
balanceAmount	Decimal	未转换免费游戏的玩家余额，或免费游戏的已转换金额
bonusRatioAmount	Decimal	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后，需要完成的总金额)
bonusMinimumConversionAmount	Decimal	免费游戏红利的最低转换金额
bonusMaximumConversionAmount	Decimal	免费游戏红利的最高转换金额
freeGameMaximumConversionAmount	Decimal	免费游戏的最高转换金额
status	Integer	免费游戏的钱包状态 0: 无效 (已取消)

		1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
expiredDate	Long	免费游戏的截止日期（以毫秒为单位的 Unix 时间戳）
updatedAt	Long	玩家被注册或重新注册到免费游戏的日期（以毫秒为单位的 Unix 时间戳）
createdDate	Long	玩家被注册到免费游戏的日期（以毫秒为单位的 Unix 时间戳）



例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "balanceId": 111261,
        "playerId": 7443,
        "playerName": "player1",
        "walletType": "B",
        "balanceAmount": 874.4,
        "bonusRatioAmount": 874.4,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdDate": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "freeGameCount": 6,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdDate": 1546918715000,
        "updatedAt": 1547028195000,
      }
    ]
  },
  "error": null
}
```

## 2.1.7 获取免费游戏的单个玩家详情:

获取免费游戏里单个玩家的详细信息:

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetPlayerFreeGames

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
player_name	String	是	免费游戏未注册玩家的唯一标识符
status	Integer Array	否	免费游戏钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&player\_name=player1&status=1&status=2

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	免费游戏未注册玩家列表

JSON 数组

参数名称	数据类型	说明
freeGameId	Integer	免费游戏的唯一标识符
freeGameName	String	免费游戏的名称
transactionId	String	交易的唯一标识符
gameIds	Integer Array	游戏的唯一标识符
gameCount	Integer	给定免费游戏的总数
totalGame	Integer	每位玩家将获得的免费游戏总数
balanceAmount	Decimal	未转换免费游戏的玩家余额, 或已转换免费游戏的已转换金额
minimumConversionAmount	Decimal	免费游戏的最低转换金额
maximumConversionAmount	Decimal	免费游戏的最高转换金额
multiplier	Integer	免费游戏的投注倍数
coinSize	Decimal	免费游戏的硬币大小
expiredDate	Long	免费游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
createdDate	Long	免费游戏的创建时间 (以毫秒为单位的 Unix 时间戳)
createdBy	String	创建免费游戏的 API 或 BackOffice 用户
updatedBy	String	更新免费游戏的 API 或 BackOffice 用户
isSupressDiscard	Boolean	允许玩家取消优惠的权限 True:不允许玩家取消优惠 False:允许玩家取消优惠

status	Integer	免费游戏钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
conversionType	Char	完成免费游戏后, 红利的转换类型 C: 现金 B: 红利

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "freeGameId": 909,
        "freeGameName": "Free Game 1",
        "gameIds": [25],
        "gameCount": 10,
        "totalGame": 5,
        "balanceAmount": 0,
        "transactionId": "12345Test",
        "minimumConversionAmount": 1,
        "maximumConversionAmount": 500,
        "multiplier": 10,
        "coinSize": 10,
        "expiredDate": 1552089599000,
        "status": 2,
        "conversionType": "B",
        "createdDate": 1498036782000,
        "createdBy": "player1",
        "updatedBy": "player1",
        "isSupressDiscard": true
      },
      {
        "freeGameId": 864,
        "freeGameName": "Free Game 2",
        "gameIds": [6],
        "gameCount": 55,
        "totalGame": 50,
        "balanceAmount": 0,
        "transactionId": "12345Test",
        "minimumConversionAmount": 1,
        "maximumConversionAmount": 500,
        "multiplier": 1,
        "coinSize": 1,
        "expiredDate": 1556672461000,
        "status": 5,
        "conversionType": "C",
        "createdDate": 1498036782000,
        "createdBy": "player1",
        "updatedBy": "player1",
        "isSupressDiscard": true
      }
    ]
  },
  "error": null
}
```

## 2.1.8 创建免费游戏

创建免费游戏：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CreateFreeGame

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	免费游戏的货币
free_game_name	String	是	免费游戏的名称
expired_date	long	是	免费游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
bonus_minimum_conversion_amount	Decimal	否*	红利的最低转换金额 空值：无转换金额限制
bonus_maximum_conversion_amount	Decimal	否*	红利的最高转换金额 空值：无转换金额限制
minimum_conversion_amount	Decimal	否*	免费游戏的最低转换金额 空值：无转换金额限制
maximum_conversion_amount	Decimal	否	免费游戏的最高转换金额 空值：无转换金额限制
bonus_type	Integer	否*	红利的下注条件 0: 红利 1: 现金 2: 红利和现金
bonus_ratio_amount**	Decimal	否*	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后, 需要完成的总金额)
bonus_ratio**	Decimal	否*	红利游戏的流水要求

			(指免费游戏里的所赢金额在进入红利钱包之后，需要完成的免费游戏投注额倍数) 投注额 = 红利比率 x 已转换的红利金额
conversion_type	Char	是	完成免费游戏后的转换类型 C: 现金 B: 红利
multiplier	Integer	是	免费游戏的投注倍数
coin_size	Decimal	是	免费游戏的硬币大小
game_count	Integer	是	每位玩家将获得的免费游戏总数
game_ids	Integer Array	是	游戏的唯一标识符 (注：只接受有着相同基本投注的游戏)
is_event	Boolean	是	免费游戏类型： True: 弹性 False: 固定
player_name	String	否	玩家的唯一标识符。系统将创建免费游戏并直接分配给该玩家。
transaction_id	String	是	交易的唯一标识符 注：对于重复的请求，系统将返回对应的免费游戏 ID

\*仅适用于红利转换类型。

\*\* 若有提供 bonus\_ratio\_amount, bonus\_ratio 的流水条件将被覆盖。

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&free_game_name=FreeGame2019&expired_date=1745289243000&status=1&bonus_maximum_conversion_amount=10&maximum_conversion_amount=1&bonus_type=1&bonus_ratio_amount=2&conversion_type=B&multiplier=5&coin_size=20&game_count=20&game_ids=1&game_ids=2&is_event=true&player_name=player1&transaction_id=12345
```

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
freeGameId	Integer	免费游戏的唯一标识符

例子:

```
{
  "data": {
    "freeGameId ": 10915
  },
  "error": null
}
```



## 2.1.9 按投注金额创建免费游戏

按照需要的投注金额创建免费游戏。

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CreateFreeGameByBetAmount

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	免费游戏的货币
free_game_name	String	是	免费游戏的名称
expired_date	long	是	免费游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
bonus_minimum_conversion_amount	Decimal	否*	红利的最低转换金额 空值: 无转换金额限制
bonus_maximum_conversion_amount	Decimal	否*	红利的最高转换金额 空值: 无转换金额限制
minimum_conversion_amount	Decimal	否	免费游戏的最低转换金额 空值: 无转换金额限制
maximum_conversion_amount	Decimal	否	免费游戏的最高转换金额 空值: 无转换金额限制
bonus_type	Integer	否*	红利的下注条件 0: 红利 1: 现金 2: 红利和现金
bonus_ratio_amount**	Decimal	否*	红利游戏的流水要求 (指免费游戏里的所赢金额在进入红利钱包之后, 需要完成的总金额)
bonus_ratio**	Decimal	否*	红利游戏的流水要求

			(指免费游戏里的所赢金额在进入红利钱包之后, 需要完成的免费游戏投注额倍数) 投注额 = 红利比率 × 已转换的红利金额
conversion_type	Char	是	完成免费游戏后的红利转换类型 C: 现金 B: 红利
bet_amount	Decimal	是	每个游戏所需的投注金额。 <b>注: 系统会自动取整至最接近的倍数和硬币大小</b>
game_count	Integer	是	每位玩家将获得的免费游戏总数
game_ids	Integer Array	是	游戏的唯一标识符 <b>(注: 只接受有着相同基本投注的游戏)</b>
is_event	Boolean	是	免费游戏类型: True: 弹性 False: 固定
player_name	String	否	玩家的唯一标识符。系统将创建免费游戏并直接分配给该玩家。
transaction_id	String	是	交易的唯一标识符。 <b>注: 对于重复的请求, 系统将返回对应的免费游戏 ID</b>

\*仅适用于红利转换类型。

\*\*若有提供 bonus\_ratio\_amount, bonus\_ratio 的流水条件将被覆盖。

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&free_game_name=FreeGame2019&expired_date=1745289243000&status=1&bonus_maximum_conversion_amount=10&maximum_conversion_amount=1&bonus_type=1&bonus_ratio_amount=2&conversion_type=B&bet_amount=0.6&game_count=20&game_ids=1&game_ids=2&is_event=true&player_name=player1&transaction_id=12345
```

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
freeGameId	Integer	免费游戏的唯一标识符

例子:

```
{
  "data": {
    "freeGameId ": 10915
  },
  "error": null
}
```

## 2.1.10 将玩家转移到固定免费游戏中

添加玩家到固定免费游戏中。

### 请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/TransferInFreeGame

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
free_game_id	Integer	是	免费游戏的唯一标识符
player_names	String Array	是	玩家的唯一标识符
transfer_reference	String	是	转移的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&free\_game\_id=12&player\_names=player1&player\_names=player2&transfer\_reference=freegame\_20181128

## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
transactionId	String	交易的唯一标识符
operatorToken	String	运营商的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
status	Integer	转移请求的状态 0: 失败 1: 成功 2: 待定（对于尚未创建的玩家，免费游戏将在创建玩家后自动转移给该玩家）

例子:

```
{
  "data": [
    {
      "transactionId": 10,
      "operatorToken": "abc123",
      "playerName": "player1",
      "currencyCode": "CNY",
      "status": 1
    },
    {
      "transactionId": 1,
      "operatorToken": "1234",
      "playerName": "player2",
      "currencyCode": "CNY",
      "status": 1
    }
  ],
  "error": null
}
```

## 2.1.11 将玩家转移到弹性免费游戏中

添加玩家到弹性免费游戏中。运营商可以覆盖特定玩家的免费游戏设置。

### 请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/TransferInFlexibleFreeGame

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_free_games	JSON String Array	是	覆盖玩家的免费游戏默认信息
free_game_id	Integer	是	免费游戏的唯一标识符
transfer_reference	String	是	转移的唯一标识符
allow_multiple	Boolean	是	允许玩家重新注册 <b>True:</b> 玩家可以重新注册免费游戏的次数 <b>False:</b> 玩家只能在免费游戏中注册一次

player\_free\_games JSON 字符串:

参数名称	数据类型	强制与否?	说明
player_name	String	是	运营商的唯一标识符
free_game_count	Integer	是	免费游戏的计数将覆盖玩家的默认计数 <b>值度: 值 ≥ 1</b>
is_unlimited_bonus_maximum_conversion_amount	Bool	是	将红利的最高转换金额设置为无限制
is_unlimited_free_game_maximum_conversion_amount	Bool	是	将免费游戏的最高转换金额设置为无限制
bonus_minimum_conversion_amount	Decimal	否*	可为玩家覆盖的最低红利转换金额 <b>值度: Value ≥ 1</b> <b>默认值: 0</b> <b>注意: 如果值 = 0, 系统将使用默认转换金额</b>
bonus_maximum_conversion_amount	Decimal	否*	可为玩家覆盖的最高红利转换金额 <b>值度: 值 ≥ 1</b> <b>默认值: 0</b> <b>注: 如果值 = 1, 系统将使用默认转换金额</b>
free_game_minimum_conversion_amount	Decimal	否*	可为玩家覆盖的最低免费游戏转换金额 <b>值度: 值 ≥ 1</b> <b>默认值: 0</b> <b>注: 如果值 = 1, 系统将使用默认转换金额</b>
free_game_maximum_conversion_amount	Decimal	否**	可为玩家覆盖的最高免费游戏转换金额 <b>值度: 值 ≥ 1</b> <b>默认值: 0</b> <b>注: 如果值 = 1, 系统将使用默认转换金额</b>
说明	String	否	转移参考说明

\*仅适用于 is\_unlimited\_bonus\_maximum\_conversion\_amount 参数值为错误的情况。

\*\*仅适用于 is\_unlimited\_free\_game\_maximum\_conversion\_amount 参数值为错误的情况。

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_free_games={"player_name":"player1","free_game_count":10,"is_unlimited_bonus_maximum_conversion_amount":false,"is_unlimited_free_game_maximum_conversion_amount":false,"bonus_maximum_conversion_amount":100.50,"free_game_maximum_conversion_amount":100.50,"description":"VIP"}&free_game_id=3&transfer_reference=freegame_20181128&allow_multiple=false
```

## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
transactionId	String	交易的唯一标识符
operatorToken	String	运营商的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
freeGameCount	Integer	玩家的免费游戏数量
status	Integer	转移请求的状态 0: 失败 1: 成功 2: 待定 (对于尚未创建的玩家, 免费游戏将在创建玩家后自动转移给该玩家)

例子:

```
{
  "data": [
    {
      "transactionId": 10,
      "operatorToken": "abc123",
      "playerName": "player1",
      "currencyCode": "CNY",
      "freeGameCount": 10,
      "status": 1
    },
    {
      "transactionId": 1,
      "operatorToken": "1234",
      "playerName": "player2",
      "currencyCode": "CNY",
      "freeGameCount": 10,
      "status": 1
    }
  ],
  "error": null
}
```

## 2.1.12 取消免费游戏

取消免费游戏：

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelFreeGame

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
free_game_id	String	是	免费游戏的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&free\_game\_id=10919

返回结果

参数：

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果： False: 失败 True: 成功

例子：

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```



### 2.1.13 取消玩家的免费游戏

取消特定玩家的免费游戏:

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPlayerFreeGame

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
balance_id	Integer	是	玩家免费游戏钱包的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player1&balance\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果: False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.1.14 取消未注册玩家的免费游戏

取消特定未注册玩家的免费游戏:

请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPendingPlayerFreeGame

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
pending_wallet_id	Integer	是	免费游戏玩家钱包的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&pending\_wallet\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果: False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.1.15 取消玩家的所有免费游戏

取消已注册/未注册玩家的所有免费游戏。

### 请求

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPlayerAllFreeGame

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
free_game_id	Integer	是	免费游戏的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player1&free\_game\_id=10919

### 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果: False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.2 红利 [最新版本: v1]

### 2.2.1 获取红利游戏列表

获取已创建的红利游戏列表:

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonus

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
from_time	Long	否	红利游戏的开始时间范围
to_time	Long	否	(以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	红利游戏的创建时间范围
to_created_time	Long	否	(以毫秒为单位的 Unix 时间戳)
bonus_id	Integer	否	红利游戏的唯一标识符
currency	String	否	红利游戏的货币
bonus_name	String	否	红利游戏的名称
bonus_transfer_type	Integer	否	红利游戏的转移类型 0: 固定 3: 弹性
status	Integer	否	红利游戏状态 0: 无效 1: 活跃 2: 已过期

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&from_time=1346592723000&to_time=1546592723000&from_created_time=1346592723000&to_created_time=1546592723000&bonus_id=12345&currency=CNY&bonus_name=BonusGame2019&bonus_transfer_type=1&status=1
```

## 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	多个红利游戏的详情列表

JSON 数组

参数名称	数据类型	说明
operatorToken	String	运营商的唯一标识符
bonusId	Integer	红利游戏的唯一标识符
bonusName	String	红利游戏的名称
isEvent	Boolean	红利游戏类型： True: 弹性 False: 固定
gameIds	Integer Array	游戏的唯一标识符
currencyCode	String	红利游戏货币
bonusRatio	Decimal	红利游戏的流水要求 指需要完成的投注额倍数
bonusRatioAmount	Decimal	红利游戏的流水要求 指需要完成的总金额
bonusMinimumConversion Amount	Decimal	免费游戏转现金的最高转换金额
bonusMaximumConversion Amount	Decimal	免费游戏转现金的最低转换金额
bonusType	Integer	红利游戏流水条件 0: 红利 1: 现金 2: 红利和现金
status	Integer	红利游戏的状态 0: 无效（已取消）

		1: 活跃 2: 已过期
convertedToCashAmount	Integer	已转换成现金的红利总额
bonusAmountGiven	Decimal	已送出的红利总额
numberOfPlayers	Integer	红利游戏中的玩家总数
expiredDate	Long	红利游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
createdDate	Long	红利游戏的创建日期 (以毫秒为单位的 Unix 时间戳)
isSuppressDiscard	Boolean	允许玩家取消优惠的权限 True: 不允许玩家取消优惠 False: 允许玩家取消优惠

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "operatorToken": "abcd",
        "bonusId": 118,
        "bonusName": "bonus118",
        "bonusType": 0,
        "isEvent": false,
        "gameIds": [
          1
        ],
        "amount": 57000,
        "currencyCode": "MYR",
        "bonusRatio": 3,
        "bonusRatioAmount": 171000,
        "bonusMinimumConversionAmount": 1000.00,
        "bonusMaximumConversionAmount": 1500.00,
        "status": 1,
        "numberOfPlayers": 0,
        "bonusAmountGiven": 0,
        "createdDate": 1497001371000,
        "isSuppressDiscard": false
      },
      {
        "operatorToken": "abcd",
        "bonusId": 119,
        "bonusName": "bonus119",
        "bonusType": 0,
        "isEvent": false,
        "gameIds": [
          1
        ],
        "amount": 500,
        "currencyCode": "USD",
        "bonusRatio": 2,
        "bonusRatioAmount": 1000,
        "status": 1,
        "numberOfPlayers": 1,
        "bonusAmountGiven": 0,
        "createdDate": 1497239059000
      }
    ]
  },
  "error": null
}
```

## 2.2.2 获取已转换为现金的**红利游戏**列表

获取已转换为现金的**红利游戏**列表:

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusConvertedWallet

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
bonus_id	Integer	是	红利的唯一标识符
player_name	String	否	玩家的唯一标识符 <b>注: 实行模糊匹配</b>
from_transaction_time	Long	否	<b>红利玩家交易的时间范围</b>
to_transaction_time	Long	否	<b>(以毫秒为单位的 Unix 时间戳)</b>

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&bonus\_id=12345&player\_name=player1&from\_transaction\_time=1346592723000&to\_transaction\_time=1346592723000

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalConvertedAmount	Decimal	已通过 <b>红利游戏</b> 转换成现金的总金额
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	<b>红利游戏已转换为现金的玩家的详细信息列表</b>



## JSON 数组

参数名称	数据类型	说明
balanceId	Integer	玩家红利游戏钱包的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
gameId	Integer	游戏的唯一标识符
transactionAmount	Decimal	玩家的转换金额
transactionDateTime	Long	转换时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": {
    "totalConvertedAmount": 10,
    "totalCount": 1,
    "totalPage": 1,
    "result": [
      {
        "balanceId": 13787,
        "playerName": "qa18",
        "currencyCode": "CNY",
        "gameId": 8,
        "transactionAmount": 10,
        "transactionDateTime": 1547449915000
      }
    ],
    [
      {
        "balanceId": 13788,
        "playerName": "qa18",
        "currencyCode": "CNY",
        "gameId": 8,
        "transactionAmount": 10,
        "transactionDateTime": 1547449915100
      }
    ]
  },
  "error": null
}
```

## 2.2.3 获取红利游戏玩家

获取已注册红利游戏玩家：

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusPlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
bonus_id	Integer	是	红利的唯一标识符
player_name	String	否	红利玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_time	Long	否	红利游戏的开始时间范围
to_time	Long	否	(以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	玩家被注册到红利游戏的时间范围
to_created_time	Long	否	(以毫秒为单位的 Unix 时间戳)
status	Integer	否	红利钱包的状态 0: 无效（已取消） 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_name=player1&from_time=1346592723000&to_time=1546592723000&from_created_time=1346592723000&to_created_time=1546592723000&bonus_id=12345&status=1
```

## 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	红利游戏的玩家列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	玩家红利钱包的唯一标识符
playerId	Integer	红利游戏玩家的唯一标识符
playerName	String	红利游戏玩家的唯一标识符
walletType	String	红利游戏玩家的钱包类型 G: 免费游戏 B: 红利
balanceAmount	Decimal	玩家的红利余额
bonusRatioAmount	Decimal	红利游戏的流水要求 指需要完成的总金额
bonusMinimumConversion Amount	Decimal	红利的最低转换金额
bonusMaximumConversion Amount	Decimal	红利的最高转换金额
status	Integer	红利钱包的状态 0: 无效（已取消） 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
expiredDate	Long	红利游戏的截止日期 (以毫秒为单位的 Unix 时间戳)

updatedAt	Long	玩家被注册或重新注册到红利游戏的日期 (以毫秒为单位的 Unix 时间戳)
createdAt	Long	

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "balanceId": 111261,
        "playerId": 7443,
        "playerName": "player1",
        "walletType": "B",
        "balanceAmount": 874.4,
        "bonusRatioAmount": 874.4,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdAt": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdAt": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

## 2.2.4 获取未注册的**红利游戏**玩家

获取待注册的**红利游戏**玩家：

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusPendingPlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
bonus_id	Integer	是	<b>红利游戏</b> 的唯一标识符
player_name	String	否	未注册 <b>红利游戏</b> 玩家的唯一标识符 <b>注：实行模糊匹配</b>
created_by	String	否	<b>注册玩家的 API 或 BackOffice 用户</b>
status	Integer	否	<b>未注册玩家的钱包状态：</b> 0: 无效（已取消） 1: <b>待处理</b> 2: 已转移

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&player\_name=player1&bonus\_id=12345&created\_by=external&status=1

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	未注册的 <b>红利游戏</b> 玩家的列表

JSON 数组

参数名称	数据类型	说明
pendingWalletId	Integer	<b>未注册玩家的红利钱包的唯一标识符</b>
playerName	String	未注册的 <b>红利游戏</b> 玩家的唯一标识符
status	Integer	<b>未注册玩家的钱包状态</b> 0: 无效 (已取消) 1: <b>待处理</b> 2: <b>已转换</b>
updatedAt	Long	<b>玩家被注册或重新注册到红利游戏的日期</b>
createdAt	Long	<b>(以毫秒为单位的 Unix 时间戳)</b>
createdBy	String	<b>注册玩家的 API 或 BackOffice 用户</b>
updatedAt	String	

例子:

```
{
  "data": {
    "totalCount": 2,
    "totalPage": 1,
    "result": [
      {
        "pendingWalletId": 140392,
        "playerName": "eddywang9999",
        "status": 1,
        "createdDate": 1554351552000,
        "updatedAt": 1554351552000,
        "createdBy": "External API"
      },
      {
        "pendingWalletId": 140393,
        "playerName": "eddywang99999",
        "status": 1,
        "createdDate": 1554351682000,
        "updatedAt": 1554351682000,
        "createdBy": "External API"
      }
    ]
  },
  "error": null
}
```

## 2.2.5 获取红利游戏的所有玩家详情

获取红利游戏的所有玩家的详细信息：

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusAllPlayers

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
bonus_id	Integer	是	红利游戏的唯一标识符
player_name	String	否	红利游戏玩家的唯一标识符 <b>注：实行模糊匹配</b>
from_time	Long	否	红利游戏的开始时间范围
to_time	Long	否	(以毫秒为单位的 Unix 时间戳)
from_created_time	Long	否	红利游戏的创建时间范围
to_created_time	Long	否	(以毫秒为单位的 Unix 时间戳)
status	Integer	否	红利钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_name=player1&from_time=1346592723000&to_time=1546592723000&from_created_time=1346592723000&to_created_time=1546592723000&bonus_id=12345&status=1
```



## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	所有红利游戏玩家的列表

JSON 数组

参数名称	数据类型	说明
balanceId	Integer	玩家红利钱包的唯一标识符
playerId	Integer	红利游戏玩家的唯一标识符
playerName	String	红利游戏玩家的唯一标识符
walletType	String	玩家的钱包类型 G: 免费游戏 B: 红利
balanceAmount	Decimal	未转换红利游戏的玩家余额, 或红利游戏的已转换金额
bonusRatioAmount	Decimal	红利游戏的流水要求 指需要完成的总金额
bonusMinimumConversionAmount	Decimal	红利游戏的最低转换金额
bonusMaximumConversionAmount	Decimal	红利游戏的最高转换金额
status	Integer	红利钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
expiredDate	Long	红利游戏的截止日期 (以毫秒为单位的 Unix 时间戳)

updatedAt	Long	玩家被注册或重新注册到红利游戏的日期
createdAt	Long	(以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "balanceId": 111261,
        "playerId": 7443,
        "playerName": "player1",
        "walletType": "B",
        "balanceAmount": 874.4,
        "bonusRatioAmount": 874.4,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdAt": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdAt": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

## 2.2.6 获取红利游戏的单个玩家详情

获取红利游戏里单个玩家的详细信息：

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/GetPlayerBonus

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
player_name	String	是	红利玩家的唯一标识符
status	Integer Array	否	红利钱包的状态 0: 无效（已取消） 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&page\_number=1&row\_count=500&player\_name=player1&status=1&status=1&status=2

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	单个玩家的红利游戏列表

JSON 数组

参数名称	数据类型	说明
bonusId	Integer	红利游戏的唯一标识符 <b>注: 对于从免费游戏转换而来的红利, 系统将返还免费游戏 ID</b>
bonusName	String	红利游戏的名称
gameIds	Integer Array	游戏的唯一标识符
balanceAmount	Decimal	未转换红利游戏的玩家余额, 或红利游戏的已转换金额
maximumConversionAmount	Decimal	红利的最高转换金额
bonusRatioAmount	Decimal	红利游戏的流水要求 指需要完成的总金额
bonusParentType	Integer	母红利类型: B: 正常创建的红利游戏 G: 从免费游戏转换而来的红利游戏
status	Integer	红利钱包的状态 0: 无效 (已取消) 1: 活跃 2: 已过期 3: 已转换 4: 已完成 5: 全新 6: 已取消用户 7: 已锁定 8: 待处理
createdDate	Long	创建红利游戏的日期

		(以毫秒为单位的 Unix 时间戳)
expiredDate	Long	红利游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
createdBy	String	创建或更新红利游戏的 API 或 BackOffice 用户
updatedBy	String	

例子:

```
{
  "data": {
    "totalCount": 2583,
    "totalPage": 259,
    "result": [
      {
        "bonusId": 11385,
        "bonusName": "Bonus 1",
        "bonusParentType": "B",
        "gameIds": [
          9
        ],
        "balanceAmount": 1000000,
        "bonusRatioAmount": 1000,
        "maximumConversionAmount": 5000,
        "status": 2,
        "createdDate": 1531981770000,
        "expiredDate": 1532015999000,
        "createdBy": "Operator1",
        "updatedBy": "Operator1"
      },
      {
        "bonusId": 12271,
        "bonusName": "Bonus 2",
        "bonusParentType": "G",
        "gameIds": [
          27
        ],
        "balanceAmount": 1000000,
        "bonusRatioAmount": 10000,
        "maximumConversionAmount": 500,
        "status": 2,
        "createdDate": 1537413312000,
        "expiredDate": 1537487999000,
        "createdBy": "Operator1",
        "updatedBy": "Operator1"
      }
    ]
  },
  "error": null
}
```

## 2.2.7 创建红利游戏

创建新的红利游戏：

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/CreateBonus

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	红利游戏货币
bonus_name	String	是	红利游戏的名称
bonus_amount	decimal	是	给予每个玩家的红利金额
expired_date	long	是	红利游戏的截止日期 (以毫秒为单位的 Unix 时间戳)
minimum_conversion_amount	Decimal	No	红利游戏的最低转换金额 空值：无转换金额限制
maximum_conversion_amount	Decimal	否	红利游戏的最高转换金额 空值：无转换金额限制
bonus_type	Integer	是	红利游戏的流水条件 0: 红利 1: 现金 2: 红利和现金
bonus_ratio_amount**	Decimal	否	红利游戏的流水要求 指需要完成的投注额倍数
bonus_ratio**	Decimal	是	红利游戏的流水要求 指需要完成的总金额 投注额 = 红利比率 × 已转换的红利金额
game_ids	Integer Array	是	游戏的唯一标识符 注：只接受有着相同基本投注的游戏

is_event	Boolean	是	<b>红利游戏类型:</b> True: 弹性 False: 固定
player_name	String	否	玩家的唯一标识符。系统将创建并直接分配红利游戏给该玩家。
transaction_id	String	是	交易的唯一标识符 <b>注: 对于重复的请求, 系统将返回对应的免费游戏 ID</b>

\*\* 若有提供 *bonus\_ratio\_amount*, *bonus\_ratio* 的流水条件将被覆盖。

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&bonus_name=Bonus
2019&bonus_amount=100.00&expired_date=1745289243000&status=1&maximum_conversion_
amount=1&bonus_type=1&bonus_ratio_amount=2&game_ids=1&game_ids=2&is_event=true&playe
r_name=player1&transaction_id=12345
```

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
bonusId	Integer	红利的唯一标识符

例子:

```
{
  "data": {
    "bonusId ": 10915
  },
  "error": null
}
```

## 2.2.8 将玩家转移到固定红利游戏中

将玩家转移到已创建的固定红利游戏中。

运营商只能将玩家转移到预先设置的红利游戏中。

### 请求

API URL : {PgSoftAPIDomain}/Bonus/v1/TransferInBonus

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
bonus_id	Integer	是	红利游戏的唯一标识符
player_names	String Array	是	玩家的唯一标识符
transfer_reference	String	是	转移的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&bonus\_id=12&player\_names=player1&player\_names=player2&transfer\_reference=bonusgame\_20181128

### 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
transactionId	String	交易的唯一标识符
operatorToken	String	运营商的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
status	Integer	转移请求的状态 0: 失败 1: 成功 2: 待处理 (对于尚未创建的玩家, 红利游戏将在创建玩家后自动分配给该玩家)



例子:

```
{
  "data": [
    {
      "transactionId": 10,
      "operatorToken": "abc123",
      "playerName": "player1",
      "currencyCode": "CNY",
      "status": 1
    },
    {
      "transactionId": 1,
      "operatorToken": "1234",
      "playerName": "player2",
      "currencyCode": "CNY",
      "status": 1
    }
  ],
  "error": null
}
```

## 2.2.9 将玩家转移到弹性红利游戏中

将玩家转移到已创建的弹性红利游戏中。

运营商可以覆盖特定玩家的红利游戏设置。

### 请求

API URL : {PgSoftAPIDomain}/Bonus/v1/TransferInFlexibleBonus

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_bonuses	JSON String Array	是	覆盖玩家的默认红利游戏信息
bonus_id	Integer	是	红利游戏的唯一标识符
transfer_reference	String	是	转移的唯一标识符
allow_multiple	Boolean	是	<b>True:</b> 玩家可以重新注册红利游戏的次数 <b>False:</b> 玩家只能在红利游戏中注册一次

player\_bonuses 字符串:

参数名称	数据类型	强制与否?	说明
player_name	String	是	运营商的唯一标识符
bonus_amount	Integer	是	玩家的自定义红利金额 <b>值度: 值≥1</b>
is_unlimited_bonus_maximum_conversion_amount	Boolean	是	将红利游戏最高转换金额设置为无限制
bonus_minimum_conversion_amount	Decimal	否*	玩家自定义的红利游戏最低转换金额 <b>值度: 值≥1</b> <b>默认值: 0</b>

			注：如果值 = 0，系统将使用默认转换金额
bonus_maximum_conversion_amount	Decimal	否*	玩家自定义的红利游戏最高转换金额 值度：值≥1 默认值：0 注：如果值 = 0，系统将使用默认转换金额
说明	String	否	转移的参考说明

\*仅适用于 *is\_unlimited\_bonus\_maximum\_conversion\_amount* 参数值为错误的情况。

例子：

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_bonuses={"player_name":
"player1","bonus_amount":10,"is_unlimited_bonus_maximum_conversion_amount":false,
"bonus_maximum_conversion_amount":100.50,"description":"VIP"}&bonus_id=3&transfer_
reference=bonusgame_20181128&allow_multiple=false
```

## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
transactionId	String	交易的唯一标识符
operatorToken	String	运营商的唯一标识符
playerName	String	玩家的唯一标识符
currencyCode	String	玩家的货币
status	Integer	转移请求的状态 0: 失败 1: 成功 2: 待处理 (对于尚未创建的玩家, 红利游戏将在创建玩家后自动分配给该玩家)

例子:

```
{
  "data": [
    {
      "transactionId": 10,
      "operatorToken": "abc123",
      "playerName": "player1",
      "currencyCode": "CNY",
      "status": 1
    },
    {
      "transactionId": 1,
      "operatorToken": "1234",
      "playerName": "player2",
      "currencyCode": "CNY",
      "status": 1
    }
  ],
  "error": null
}
```

## 2.2.10 取消红利游戏

取消已创建的红利游戏:

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/CancelBonus  
HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
bonus_id	String	是	红利游戏的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&bonus\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果: False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.2.11 取消已注册玩家的红利游戏

取消特定已注册玩家的红利游戏:

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/CancelPlayerBonus

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
balance_id	Integer	是	玩家红利钱包的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player1&balance\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果: False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.2.12 取消未注册玩家的红利游戏

取消特地未注册玩家的红利游戏:

请求

API URL : {PgSoftAPIDomain}/Bonus/v1/CancelPendingPlayerBonus  
HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
pending_wallet_id	Integer	是	玩家红利钱包的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&pending\_wallet\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	请求结果 False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```

## 2.2.13 取消玩家的所有红利游戏

取消玩家的所有红利游戏:

请求

API URL : {PgSoftAPIDomain}/**Bonus**/v1/CancelPlayerAllBonus

HTTP 方法 : POST

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
bonus_id	Integer	是	红利的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player1&bonus\_id=10919

返回结果

参数:

JSON 对象

参数名称	数据类型	说明
actionResult	Integer	<b>请求结果:</b> False: 失败 True: 成功

例子:

```
{
  "data": {
    "actionResult": true
  },
  "error": null
}
```



## 2.3 锦标赛 [最新版本: v2]

### 2.3.1 获取锦标赛

获取锦标赛列表。

请求

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentsAsync

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
status	Integer	否	锦标赛状态: 0: 取消 1: 进行中 2: 全新 3: 结束
game_ids	Integer Array	否	游戏的唯一标识符
language	String	否	锦标赛信息的相应语言: en-US: 英语 (默认) zh-CN: 中文
start_time	Long	否	锦标赛开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳)
end_time	Long	否	

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&status=1&game\_ids=1&game\_ids=2&language=zh-cn&start\_time=1559145500000&end\_time=1559145600000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
tournamentKey	String	锦标赛的唯一键
tournamentId	Long	锦标赛的唯一标识符
subTournamentId	Long	副锦标赛的唯一标识符
name	String	锦标赛的名称
description	String	锦标赛的说明
registrationFees	String	锦标赛的注册费
prize	String	锦标赛的奖励
registrationUrl	String	锦标赛的注册 URL
hasSubTournaments	Boolean	说明副锦标赛是否存在
subTournamentStatus	Decimal	副锦标赛的状态
subTournamentStartTime	Long	副锦标赛的开始时间 (以毫秒为单位的 Unix 时间戳)
subTournamentEndTime	Long	副锦标赛的结束时间 (以毫秒为单位的 Unix 时间戳)
numberOfPlayers	Integer	锦标赛的参赛人数
initialPoint	Decimal	锦标赛的初始余额
qualifiedPoint	Decimal	锦标赛玩家的排名资格
pointRatio	Decimal	进入排名所需的最低消耗点数 <b>注：仅适用于累积模式</b>
gameMode	Integer	锦标赛模式： 1: 累积模式 2: 消耗模式 3: 混合模式 (累积 + 消耗)
rankStatus	Integer	锦标赛的排名状态 0: 禁用 1: 启用

		2: 停止（锦标赛排名已完成）
status	Integer	锦标赛状态： 0: 取消 1: 进行中 2: 全新 3: 结束
startTime	Long	锦标赛的开始时间 （以毫秒为单位的 Unix 时间戳）
endTime	Long	锦标赛的结束时间 （以毫秒为单位的 Unix 时间戳）
createdTime	Long	锦标赛信息的创建或更新时间 （以毫秒为单位的 Unix 时间戳）
updatedAt	Long	

例子:

```
{
  "data": [{
    "tournamentKey": "202-99-0_P",
    "tournamentId": 202,
    "name": "Wild Inferno Tournament",
    "description": "",
    "gameIds": [52],
    "registrationFees": "",
    "prize": "Free Smile",
    "registrationUrl": "",
    "subTournamentId": 99,
    "hasSubTournaments": false,
    "numberOfPlayers": 26,
    "initialPoint": 1000,
    "qualifiedPoint": 1000.01,
    "pointRatio": 10,
    "gameMode": 1,
    "rankStatus": 0,
    "status": 1,
    "startTime": 1559145600000,
    "endTime": 1577807999000,
    "subTournamentStatus": 0,
    "subTournamentStartTime": 1559145600000,
    "subTournamentEndTime": 1559145700000,
    "createdTime": 1559145600000,
    "updatedAt": 1559145600000
  }, {
    "tournamentKey": "202-991-0_P",
    "tournamentId": 203,
    "name": "Wild Inferno Tournament",
    "description": "",
    "gameIds": [52],
    "registrationFees": "",
    "prize": "Free Smile",
    "registrationUrl": "",
    "subTournamentId": 99,
    "hasSubTournaments": false,
    "numberOfPlayers": 26,
    "initialPoint": 1000,
    "qualifiedPoint": 1000.01,
    "pointRatio": 10,
    "gameMode": 1,
    "rankStatus": 0,
    "status": 1,
    "startTime": 1559145600000,
    "endTime": 1577807999000,
    "subTournamentStatus": 0,
    "subTournamentStartTime": 1559145600000,
    "subTournamentEndTime": 1559145700000,
    "createdTime": 1559145600000,
    "updatedAt": 1559145600000
  }],
  "error": null
}
```

## 2.3.2 获取锦标赛玩家

获取锦标赛的玩家 列表。

请求

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentPlayersAsync

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
sub_tournament_id	Integer	是	副锦标赛的唯一标识符
page_number	Integer	是	返回数据的批号
row_count	Integer	是	个别批次的记录数量
player_names	String	否	锦标赛玩家的唯一标识符
from_created_time	Long	否	玩家被添加到锦标赛的时间范围。
to_created_time	Long	否	(以毫秒为单位的 Unix 时间戳)

例子:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=1000&
sub_tournament_id=1460&player_names=qafmcny18&from_created_time=1559145500000&to_c
reated_time=1559145600000
```

## 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
totalCount	Integer	记录总数
totalPage	Integer	可用批次的总数
result	JSON Array	锦标赛的玩家列表

JSON 数组

参数名称	数据类型	说明
operatorToken	String	运营商的唯一标识符
tournamentPlayerId	Long	锦标赛玩家的唯一标识符
playerName	String	玩家的唯一标识符
balanceAmount	Decimal	锦标赛玩家的余额
point	Decimal	锦标赛玩家目前的获胜点
remainingRevive	Integer	剩余的锦标赛重玩次数
remainingRegister	Integer	剩余的锦标赛注册次数
createdTime	Long	锦标赛玩家信息的创建时间 (以毫秒为单位的 Unix 时间戳)
updatedTime	Long	锦标赛玩家信息的更新时间 (以毫秒为单位的 Unix 时间戳)
createdBy	Integer	将玩家注册到锦标赛中的 API 或 BackOffice 用户

例子:

```
{
  "data": {
    "totalCount": 1,
    "result": [
      {
        "tournamentPlayerId": 193521,
        "operatorToken": "1234",
        "playerName": "qafmcny18",
        "balanceAmount": 96,
        "point": 544.05,
        "remainingRevive": 0,
        "remainingRegister": 9,
        "createdTime": 1561640502000,
        "updatedTime": 1561640502000,
        "createdBy": "mei"
      },
      {
        "tournamentPlayerId": 193522,
        "operatorToken": "1234",
        "playerName": "qafmcny19",
        "balanceAmount": 96,
        "point": 544.05,
        "remainingRevive": 0,
        "remainingRegister": 9,
        "createdTime": 1561640502000,
        "updatedTime": 1561640502000,
        "createdBy": "mei"
      }
    ],
    "totalPage": 1
  },
  "error": null
}
```

### 2.3.3 获取锦标赛的最高排名

获取锦标赛的最高排名：

请求

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentTopRankingsAsync

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
tournament_id	Long	是	锦标赛的唯一标识符
count	Integer	是	排名记录数
player_name	String	否	玩家的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&tournament\_id=41&count=10&player\_name=player1



## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
top_ranking	JSON Array	锦标赛最高排名列表

JSON 数组

参数名称	数据类型	说明
operatorToken	Integer	运营商的唯一标识符
tournamentPlayerId	Integer	锦标赛玩家的唯一标识符
playerName	String	玩家的唯一标识符
point	Decimal	玩家的当前锦标赛点数
totalSpendPoint	Decimal	玩家的总支出点数
previousRank	Integer	玩家先前的排名
rank	Integer	玩家当前的排名
lastUpdatedOn	Long	最后更新排名记录的时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data":{
    "top_ranking": [
      {
        "tournamentPlayerId": 0,
        "operatorToken": "1234",
        "playerName": "player1",
        "point": 30360.98,
        "totalSpendPoint": 69376.8,
        "previousRank": 1,
        "rank": 1,
        "lastUpdatedOn": 1540380199000
      },
      {
        "tournamentPlayerId": 0,
        "operatorToken": "1234",
        "playerName": "player2",
        "point": 10000.74,
        "totalSpendPoint": 33,
        "previousRank": 2,
        "rank": 2,
        "lastUpdatedOn": 1540380199000
      }
    ]
  },
  "error": null
}
```

## 2.3.4 注册锦标赛玩家

将多个玩家添加到锦标赛游戏中：

请求

API URL : {PgSoftAPIDomain}/Tournament/v2/CreateTournamentPlayersAsync

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
tournament_id	Long	是	锦标赛的唯一标识符 注：请使用 PG BackOffice 来创建锦标赛
player_names	String Array	是	玩家的唯一标识符
allow_re-register	Boolean	是	玩家可重新注册同一个锦标赛 True: 允许 False: 不允许

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&tournament\_id=41&player\_names=player1 &player\_names=player2&allow\_re-register=true

返回结果

参数：

JSON 数组

参数名称	数据类型	说明
operatorToken	Integer	运营商的唯一标识符
tournamentPlayerId	Integer	锦标赛玩家的唯一标识符
playerName	String	玩家的唯一标识符
balanceAmount	Integer	玩家的初始点
registerAllowed	Integer	可重新注册锦标赛的次数
reviveAllowed	Integer	可在锦标赛中复活的次数

status	Integer	锦标赛状态 0: 取消 1: 进行中 2: 全新 3: 结束
--------	---------	--

例子:

```
{
  "data": [
    {
      "tournamentPlayerId": 124,
      "operatorToken": "1234",
      "playerName": " player1",
      "balanceAmount": 10000,
      "registerAllowed": 98,
      "reviveAllowed": 3,
      "status": 1
    },
    {
      "tournamentPlayerId": 125,
      "operatorToken": "1234",
      "playerName": " player1",
      "balanceAmount": 10000,
      "registerAllowed": 98,
      "reviveAllowed": 3,
      "status": 1
    },
    {
      "tournamentPlayerId": 126,
      "operatorToken": "1234",
      "playerName": " player2",
      "balanceAmount": 10000,
      "registerAllowed": 98,
      "reviveAllowed": 3,
      "status": 1
    }
  ],
  "error": null
}
```

### 2.3.5 获取锦标赛的现金奖励列表

获取锦标赛的现金奖励列表：

请求

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentAutoRewardCashPrize

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
start_time	Long	是	记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值为 1 - 14 天</b>
end_time	Long	是	
transaction_id	Long	否	奖励交易的唯一标识符
tournament_id	Integer	否	锦标赛的唯一标识符
player_name	String	否	锦标赛玩家的唯一标识符
currency	String	否	锦标赛货币
page_number	Integer	否	返回数据的批号 默认值： 1
row_count	Integer	否	个别批次的记录数量 默认值： 5000

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&start\_time=1559145500000&end\_time=1559145600000

返回结果

参数:

JSON 数组

参数名称	数据类型	说明
totalCount	Integer	锦标赛记录总数
totalPage	Integer	锦标赛结果总页数
result	JSON Array	锦标赛的优胜者列表

result JSON 数组

参数名称	数据类型	说明
tournamentId	Integer	锦标赛的唯一标识符
currency	string	锦标赛货币
winners	JSON Array	锦标赛的优胜者列表

winners JSON 数组

参数名称	数据类型	说明
transactionId	Long	奖励交易的唯一标识符
transactionTime	Long	奖励的交易时间 (以毫秒为单位的 Unix 时间戳)
tournamentPlayerId	Integer	锦标赛玩家的唯一标识符
playerName	String	玩家的唯一标识符
rank	Integer	锦标赛的玩家排名
adjustmentAmount	Decimal	玩家在锦标赛里的奖励金额

例子:

```
{
  "data": {
    "totalCount": 3,
    "result": [
      {
        "tournamentId": 175,
        "currency": "CNY",
        "winners": [
          {
            "transactionId": 27021004,
            "txntime": 1599811942000,
            "tournamentPlayerId": 28396,
            "playerName": "calven77407A5275",
            "rank": 2,
            "adjustmentAmount": 20000.00000
          }
        ]
      },
      {
        "tournamentId": 184,
        "currency": "CNY",
        "winners": [
          {
            "transactionId": 27021007,
            "txntime": 1599811943000,
            "tournamentPlayerId": 15862,
            "playerName": "JiaXiangHong",
            "rank": 1,
            "adjustmentAmount": 200.00000
          },
          {
            "transactionId": 27021008,
            "txntime": 1599811943000,
            "tournamentPlayerId": 15669,
            "playerName": "dong",
            "rank": 2,
            "adjustmentAmount": 100.00000
          }
        ]
      }
    ]
  },
  "totalPage": 1
},
"error": null
}
```

## 2.4 投注历史记录 [最新版本: v3]

为获取各种类型的投注历史数据，PG Soft 提供了各种不同的外部应用编程接口(API)。请注意，API 数据仅在每次旋转之后的 3 – 5 分钟之内可用。此历史记录仅提供状态为“已完成”的投注记录，运营商可通过 PG 后台查看待处理和失败投注。

**注意：**

外部投注历史的 API 为专用 API，并使用独立的 URL: **{DataGrabAPIDomain}**



## 2.4.1 获取历史记录

获取投注历史记录：

### 注意

- 运营商可获得最近 60 天的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。
- PG 或将通过更新数据行版本来重新推送数据。运营商需通过 *betId* 来识别和更新受重复数据。
- 数据行版本并不是唯一值。运营商必须为每个请求提取至少 1500 条记录。

### 步骤

- **步骤一：**在第一个 GetHistory API 调用中，设置 `row_version = 1`
- **步骤二：**保存每个获取数据的请求调用中的 `rowVersion` 最大值
- **步骤三：**在接下来的调用中，将 `row_version` 值设置为步骤二里所保存的 `rowVersion`
- 重复第二及第三步骤，直到返回的记录少于所需数量（例如：每个请求 1500 条记录）
- 如果返回的记录数少于所需数量，请停止并等待一个时间间隔（建议 5 分钟）再进行下一个 API 调用
- 可通过检查每个调用中重复的 `betId` 来识别重复的记录

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetHistory  
HTTP 方法 : POST

### 参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
count	Integer	是	每批的记录数 <b>注: 值度 1500 - 5000</b>
bet_type	Integer	是	投注类型的投注记录: 1: 真实游戏
row_version	Long	是	数据更新时间 (以毫秒为单位的 Unix 时间戳) <b>注:</b> <ul style="list-style-type: none"> <li>首次呼叫时将值设置为 1</li> <li>运营商需保存每个呼叫最大的 row_version 以用作下一个呼叫请求的 row_version 值</li> </ul>
hands_status	Long	否	投注状态: 0:全部 (默认) 1: 非最后一手投注 2: 最后一手投注 3: 已调整

### 例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&bet\_type=1&row\_version=1346592723000&hands\_status=0

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	Integer	母注单的唯一标识符
betId	Integer	子投注的唯一标识符（唯一键值）
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
betAmount	Decimal*	玩家的投注额
winAmount	Decimal*	玩家的所赢金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
balanceBefore	Decimal*	玩家交易前的余额
balanceAfter	Decimal*	玩家交易后的余额
handsStatus	Integer	投注状态： 1: 非最后一手投注 2: 最后一手投注 3: 已调整
rowVersion	Integer	数据更新时间 (以毫秒为单位的 Unix 时间戳)
betTime	Integer	当前投注的开始时间

		(以毫秒为单位的 Unix 时间戳)
betEndTime	Integer	当前投注的结束时间 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位数。

例子：

```
{
  "data": [
    {
      "betId": 35677059,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 3000,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88081189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546511407,
      "betTime": 1529546613715,
      "betEndTime": 1529546700260
    },
    {
      "betId": 35677061,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 0,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88078189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546556399,
      "betTime": 1529546647867,
      "betEndTime": 1529546701360
    }
  ],
  "error": null
}
```

## 2.4.2 获取特定时间内的历史记录

获取特定时间段内的投注历史记录：

### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。

### 步骤

- **步骤一：** 在第一个 GetHistory API 调用中，将 from\_time 设置为您要的时间戳，并将 to\_time 设置为当前时间
- **步骤二：** 保存每个获取数据的请求调用中的 betEndTime 最大值
- **步骤三：** 在接下来的调用中，将 from\_time 参数值设置为步骤二里所保存的 betEndTime，to\_time 参数值可保持不变（当前时间）
- 重复第二及第三步骤，直到返回的记录少于所需数量（例如：每个请求 1500 条记录）
- 如果返回的记录数少于所需数量，请停止并等待一个时间间隔（建议 5 分钟）再进行下一个 API 调用

可通过检查每个调用中重复的 betId 来识别重复的记录

## 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetHistoryForSpecificTimeRange  
 HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
count	Integer	是	每批的记录数 <b>注: 值度 1500 - 5000</b>
bet_type	Integer	是	投注类型的投注记录: 1: 真实游戏
from_time	Long	是	投注记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳)
to_time	Long	是	
			<b>注: 值度为 1 - 40 天</b>

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&bet\_type=1&from\_time=1346592723000&to\_time=1546592723000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	Integer	母注单的唯一标识符
betId	Integer	子投注的唯一标识符（唯一键值）
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录的类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
betAmount	Decimal*	玩家的投注额
winAmount	Decimal*	玩家的所赢金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
balanceBefore	Decimal*	玩家交易前的余额
balanceAfter	Decimal*	玩家交易后的余额
handsStatus	Integer	投注状态： 1: 非最后一手投注 2: 最后一手投注 3: 已调整
rowVersion	Integer	数据更新时间 (以毫秒为单位的 Unix 时间戳)
betTime	Integer	当前投注的开始时间 (以毫秒为单位的 Unix 时间戳)
betEndTime	Integer	当前投注的结束时间 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位。

例子:

```
{
  "data": [
    {
      "betId": 35677059,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 3000,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88081189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546511407,
      "betTime": 1529546613715,
      "betEndTime": 15295476224
    },
    {
      "betId": 35677061,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 0,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88078189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546556399,
      "betTime": 1529546647867,
      "betEndTime": 1529546745249
    }
  ],
  "error": null
}
```



### 2.4.3 获取单一玩家的历史记录

获取单一玩家的投注历史记录：

#### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。

#### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerHistory  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
bet_type	Integer	是	投注类型的投注记录： 1: 真实游戏
start_time	Long	是	投注记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳)
end_time	Long	是	

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player1&start\_time=1346592723000&end\_time=1546592723000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	Integer	母注单的唯一标识符
betId	Integer	子投注的唯一标识符（唯一键值）
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录的类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
betAmount	Decimal*	玩家的投注额
winAmount	Decimal*	玩家的所赢金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
balanceBefore	Decimal*	玩家交易前的余额
balanceAfter	Decimal*	玩家交易后的余额
rowVersion	Integer	数据更新时间 (以毫秒为单位的 Unix 时间戳)
betTime	Integer	当前记录日期 (以毫秒为单位的 Unix 时间戳)
handsStatus	Integer	投注状态： 1: 非最后一手投注 2: 最后一手投注 3: 已调整

\*根据游戏和余额，系统将可能返回多达 6 个小数位数。

例子:

```
{
  "data": [
    {
      "betId": 35677059,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 3000,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88081189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546511407,
      "betTime": 1529546613715
    },
    {
      "betId": 35677061,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 0,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88078189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546556399,
      "betTime": 1529546647867
    }
  ],
  "error": null
}
```

## 2.4.4 获取玩家的每日汇总

获取玩家的每日投注汇总：

### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。
- *PG 或将通过更新数据行版本来重新推送数据。运营商需通过 betId 来识别和更新受重复数据。*
- 数据行版本并不是唯一值。运营商必须为每个请求提取至少 1500 条记录。

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerDailySummary  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
count	Integer	是	每批的记录数 <b>注： 值度 1500 - 5000</b>
bet_type	Integer	是	投注记录的类别： 1: 真实游戏
row_version	Long	是	数据更新时间 (以毫秒为单位的 Unix 时间戳)
time_zone	Integer	否	时区： -5: 世界标准时间 (UTC)-5 -4: 世界标准时间 (UTC)-4 0: 世界标准时间 (UTC) (默认) 1: 世界标准时间 (UTC)+1 4: 世界标准时间 (UTC)+4 <b>7: 世界标准时间 (UTC)+7</b> 8: 世界标准时间 (UTC)+8

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&bet\_type=1&row\_version=1346592723000&time\_zone=8

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
playerName	String	玩家的唯一标识符
playerType	Integer	玩家类别： 1: 真实玩家 2: 测试玩家
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录的类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
hands	Integer	主投注计数
games	Integer	子投注计数
betAmount	Decimal*	玩家的投注额
winLossAmount	Decimal*	玩家输赢的总金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
rowVersion	Integer	数据更新时间 (以毫秒为单位的 Unix 时间戳)
summaryDate	Integer	当前记录日期 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位。

例子:

```
{
  "data": [
    {
      "playerName": " player1",
      "playerType": 1,
      "gameId": 39,
      "betType": 1,
      "transactionType": 1,
      "platform": 1,
      "currency": "EUR",
      "hands": 2,
      "games": 2,
      "betAmount": 60.05,
      "winLossAmount": -60.05,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "rowVersion": 1545971022000,
      "summaryDate": 1545955200000
    },
    {
      "playerName": "player2",
      "playerType": 1,
      "gameId": 24,
      "betType": 1,
      "transactionType": 1,
      "platform": 1,
      "currency": "EUR",
      "hands": 9,
      "games": 9,
      "betAmount": 0.09,
      "winLossAmount": 1.91,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "rowVersion": 1546238743000,
      "summaryDate": 1546214400000
    }
  ],
  "error": null
}
```

## 2.4.5 获取玩家在特定时间内的每日汇总

获取玩家在特定时间段内的每日投注汇总：

### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。
- PG Soft 可通过更新数据行版本来重新推送数据。运营商需通过 *playerName, gameId and summaryDate* 来识别和更新重复数据。

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerDailySummaryForSpecificTimeRange

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
count	Integer	是	每批的记录数 <b>注： 值度 1500 - 5000</b>
bet_type	Integer	是	投注记录的类别： 1: 真实游戏
row_version	Long	是	数据的更新时间 (以毫秒为单位的 Unix 时间戳)
from_time	Long	是	投注记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值度为 1 - 40 天</b>
to_time	Long	是	
time_zone	Integer	否	时区： -5: 世界标准时间 (UTC)-5 -4: 世界标准时间 (UTC)-4 0: 世界标准时间 (UTC) (默认) 1: 世界标准时间 (UTC)+1 4: 世界标准时间 (UTC)+4 8: 世界标准时间 (UTC)+8

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&bet\_type=1&from\_time=1346592723000&to\_time=1546592723000&time\_zone=8

## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
playerName	String	玩家的唯一标识符
playerType	Integer	玩家类别: 1: 真实玩家 2: 测试玩家
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录的类别: 1: 真实游戏
transactionType	Integer	交易类别: 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
hands	Integer	主投注计数
games	Integer	子投注计数
betAmount	Decimal*	玩家的投注额
winLossAmount	Decimal*	玩家输赢的总金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
rowVersion	Integer	数据更新时间 (以毫秒为单位的 Unix 时间戳)
summaryDate	Integer	当前记录日期 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位。



例子:

```
{
  "data": [
    {
      "playerName": " player1",
      "playerType": 1,
      "gameId": 39,
      "betType": 1,
      "transactionType": 1,
      "platform": 1,
      "currency": "EUR",
      "hands": 2,
      "games": 2,
      "betAmount": 60.05,
      "winLossAmount": -60.05,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "rowVersion": 1545971022000,
      "summaryDate": 1545955200000
    },
    {
      "playerName": "player2",
      "playerType": 1,
      "gameId": 24,
      "betType": 1,
      "transactionType": 1,
      "platform": 1,
      "currency": "EUR",
      "hands": 9,
      "games": 9,
      "betAmount": 0.09,
      "winLossAmount": 1.91,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "rowVersion": 1546238743000,
      "summaryDate": 1546214400000
    }
  ],
  "error": null
}
```

## 2.4.6 获取投注汇总

获取每次投注的游戏汇总报表：

### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。
- *PG 将通过更新数据行版本来重新推送数据。运营商需通过 **betId** 来识别和更新受重复数据。*
- 数据行版本并不是唯一值。运营商必须为每个请求提取至少 **1500** 条记录。

### 请求

API URL : **{DataGrabAPIDomain}**/Bet/v4/GetHandsSummary  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
count	Integer	是	每批的记录数 <b>注： 值度 1500 - 5000</b>
bet_type	Integer	是	投注类型的投注记录： 1: 真实游戏
hands_status	Long	是	投注状态： 0: 全部 1: 未完成 2: 已完成
row_version	Long	是	数据更新时间 (以毫秒为单位的 Unix 时间戳)

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&bet\_type=1&hands\_status=2&row\_version=1346592723000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	Integer	母注单的唯一标识符
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
betType	Integer	投注记录的类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参考 <a href="#">平台</a> 以获取更多信息)
currency	String	记录货币
betAmount	Decimal*	玩家的投注额
winAmount	Decimal*	玩家的所赢金额
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池金额
balanceBefore	Decimal*	玩家交易前的余额
balanceAfter	Decimal*	玩家交易后的余额
handStatus	Integer	投注状态： 1: 非最后一手投注 2: 最后一手投注 3: 已调整
rowVersion	Integer	更新数据的时间 (以毫秒为单位的 Unix 时间戳)
handStartTime	Integer	当前投注的开始时间 (以毫秒为单位的 Unix 时间戳)
handEndTime	Integer	当前投注的结束时间 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位。

例子:

```
{
  "data": [
    {
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 3000,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88081189.2,
      "balanceAfter": 88078189.2,
      "handStartTime": 1529546647867,
      "handEndTime": 1529546647867,
      "handsStatus": 1
    },
    {
      "parentBetId": 35677060,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 0,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88078189.2,
      "balanceAfter": 88078189.2,
      "rowVersion": 1529546556399,
      "handStartTime": 1529546647867,
      "handEndTime": 1529546647867,
      "handsStatus": 2
    }
  ],
  "error": null
}
```

## 2.4.7 获取每小时投注汇总

获取运营商每小时的投注记录报表：

### 注意

- 运营商可以获取最近 **60 天** 的投注记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。

### 请求

API URL: {DataGrabAPIDomain}/Bet/v4/GetHandsSummaryHourly

HTTP 方法 : POST

### 参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
from_time	Long	是	投注记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值为 1 - 40 天</b>
to_time	Long	是	
currency	String	否	记录货币
transaction_types	Integer Array	否	交易类别： 1: 现金 2: 红利 3: 免费游戏

### 例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&from\_time=156729600000&to\_time=1569801600000&currency=cny&transaction\_types=1&transaction\_types=2

### 返回结果

#### 参数：

#### JSON 数组

参数名称	数据类型	说明
dateTime	Integer	主投注的唯一标识符

totalHands	Integer	玩家的唯一标识符
currency	String	记录货币
totalBetAmount	Decimal*	总投注额
totalWinAmount	Decimal*	派彩总金额
totalPlayerWinLossAmount	Decimal*	玩家的总输赢
totalCompanyWinLossAmount	Decimal*	公司的总输赢
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏

\*根据游戏和余额，系统将可能返回多达 6 个小数位数。

例子：

```
{
  "data": [
    {
      "dateTime": 1567738800000,
      "totalHands": 9,
      "currency": "CNY",
      "totalBetAmount": 25.5,
      "totalWinAmount": 64.41,
      "totalPlayerWinLossAmount": 38.91,
      "totalCompanyWinLossAmount": -38.91,
      "transactionType": 3
    },
    {
      "dateTime": 1567742400000,
      "totalHands": 41,
      "currency": "CNY",
      "totalBetAmount": 15.5,
      "totalWinAmount": 4.01,
      "totalPlayerWinLossAmount": -11.49,
      "totalCompanyWinLossAmount": -11.49,
      "transactionType": 3
    },
    {
      "dateTime": 1567746000000,
      "totalHands": 14,
      "currency": "CNY",
      "totalBetAmount": 27.0,
      "totalWinAmount": 47.95,
      "totalPlayerWinLossAmount": 20.95,
      "totalCompanyWinLossAmount": -20.95,
      "transactionType": 3
    }
  ],
  "error": null
}
```

## 2.4.8 获取玩家的投注总和

获取玩家在特定时间范围内的投注总和：

### 注意

- 运营商可获得最近 **60 天** 的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerBetsSummationTotal  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
bet_type	Integer	是	投注类型的投注记录： 1: 真实游戏
from_time	Long	是	投注记录开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值为 1 - 40 天</b>
to_time	Long	是	
currency	String	否	记录货币
game_ids	Integer Array	否	游戏的唯一标识符
player_modes	Integer Array	否	游戏模式： 0: 固定 1: 现金红利
player_type	Integer	否	玩家类别： <b>0: 所有</b> 1: 真实玩家 2: 测试玩家
transaction_type	Integer	否	交易类别： <b>0: 所有</b> 1: 现金 2: 红利 3: 免费游戏

time_zone	Integer	否	时区： -5: 世界标准时间 (UTC)-5 -4: 世界标准时间 (UTC)-4 0: 世界标准时间 (UTC) (默认) 1: 世界标准时间 (UTC)+1 4: 世界标准时间 (UTC)+4 8: 世界标准时间 (UTC)+8
-----------	---------	---	--

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&bet\_type=1&from\_time=1346592723000&to\_time=1346592723000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
currency	String	记录货币
totalGames	Integer	游戏回合总计数
totalHands	Integer	游戏投注总计数
totalBetAmountConverted	Decimal*	已转换的玩家总投注金额 (货币为欧元)
totalWinLossAmountConverted	Decimal*	已转换的玩家总输赢金额 (货币为欧元)
totalCompanyWinLossAmountConverted	Decimal*	已转换的公司总输赢金额 (货币为欧元)
totalJackpotRtpContributionAmountConverted	Decimal*	已转换的总奖池返回率贡献额记录 (货币为欧元)
totalJackpotContributionAmountConverted	Decimal*	已转换的总奖池贡献额记录 (货币为欧元)
totalJackpotWinAmountConverted	Decimal*	已转换的总奖池赢额记录 (货币为欧元)
totalBetAmount	Decimal*	总投注金额记录
totalWinLossAmount	Decimal*	总输赢金额记录
totalCompanyWinLossAmount	Decimal*	公司的总输赢金额记录



totalJackpotRtpContributionAmount	Decimal*	总奖池返回率贡献额记录
totalJackpotContributionAmount	Decimal*	总奖池贡献额记录
totalJackpotWinAmount	Decimal*	总奖池赢额记录

\*根据游戏和余额，系统将可能返回多达 6 个小数位数。

例子：

```
{
  "data": [
    {
      "currency": "CNY",
      "totalGames": 1232191,
      "totalHands": 1171198,
      "totalBetAmountConverted": 436315.92259,
      "totalWinLossAmountConverted": 476402.99206,
      "totalCompanyWinLossAmountConverted": -476402.99206,
      "totalJackpotRtpContributionAmountConverted": 0.01824,
      "totalJackpotContributionAmountConverted": 0.66614,
      "totalJackpotWinAmountConverted": 32000.13301,
      "totalBetAmount": 3417984.61,
      "totalWinLossAmount": 3732015.25799,
      "totalCompanyWinLossAmount": -3732015.25799,
      "totalJackpotRtpContributionAmount": 0.14268,
      "totalJackpotContributionAmount": 5.21856,
      "totalJackpotWinAmount": 250680.69
    },
    {
      "currency": "EUR",
      "totalGames": 75,
      "totalHands": 29,
      "totalBetAmountConverted": 266.0,
      "totalWinLossAmountConverted": 1439.8,
      "totalCompanyWinLossAmountConverted": -1439.8,
      "totalJackpotRtpContributionAmountConverted": 0.0,
      "totalJackpotContributionAmountConverted": 0.0,
      "totalJackpotWinAmountConverted": 0.0,
      "totalBetAmount": 266.0,
      "totalWinLossAmount": 1439.8,
      "totalCompanyWinLossAmount": -1439.8,
      "totalJackpotRtpContributionAmount": 0.0,
      "totalJackpotContributionAmount": 0.0,
      "totalJackpotWinAmount": 0.0
    }
  ],
  "error": null
}
```

## 2.4.9 获取玩家未完成的投注

获取玩家未完成的投注：

### 注意

- 运营商可获得最近 60 天的投注历史记录。
- 此 API 仅适用于报告目的，我们不建议您使用此 API 于任何其他业务操作。若您需要使用此 API 于业务运营，请联系 PG。

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerUnfinishedGamesBet

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符
from_time	Long	是	锦标赛开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值为 1 - 7 天</b>
to_time	Long	是	
game_ids	Integer Array	否	游戏的唯一标识符
player_modes	Integer Array	否	游戏模式： 0: 固定 1: 现金红利
player_type	Integer	否	玩家类别： 0: 所有 1: 真实玩家 2: 测试玩家
transaction_type	Integer	否	交易类别： 0: 所有 1: 现金 2: 红利

			3: 免费游戏
time_zone	Integer	否	时区： -5: 世界标准时间 (UTC)-5 -4: 世界标准时间 (UTC)-4 0: 世界标准时间 (UTC) (默认) 1: 世界标准时间 (UTC)+1 4: 世界标准时间 (UTC)+4 8: 世界标准时间 (UTC)+8

例子：

operator\_token=abcd&secret\_key=abc123&game\_ids=1&player\_name=player123&date\_from=1588723200000&date\_to=1588982400000

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	String	母注单的唯一标识符
gameId	Integer	游戏的唯一标识符
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台
currency	String	记录货币
betAmount	Decimal	玩家投注
winAmount	Decimal	玩家派彩
jackpotContributionAmount	Decimal	玩家的奖池贡献额
balanceBefore	Decimal	交易前的玩家余额
balanceAfter	Decimal	交易后的玩家余额
handStartTime	Long	投注的开始时间 (以毫秒为单位的 Unix 时间戳)
handEndTime	Long	投注的结束时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": [
    {
      "parentBetId": "35677059",
      "gameId": 69,
      "transactionType": 1,
      "platform": 1,
      "currency": "CNY",
      "betAmount": 200.0,
      "winAmount": 4840.0,
      "jackpotContributionAmount": 0.0,
      "balanceBefore": 1031747.0,
      "balanceAfter": 1036387.0,
      "handStartTime": 1599631083541,
      "handEndTime": 1599631097033,
    },
  ],
  "error": null
}
```

## 2.4.10 获取投注详情

检索多个投注的详细信息：

### 注意

- 运营商可以检索最近 60 天的投注历史记录。

### 请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetBetDetails

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
bet_ids	Integer	是	投注的唯一标识符

例子：

operator\_token=abcd&secret\_key=abc123&bet\_ids=1315486539764989952&bet\_ids=1315486539764989960

## 返回结果

参数：

JSON 数组

参数名称	数据类型	说明
parentBetId	Integer	母注单的唯一标识符
betId	Integer	子投注的唯一标识符（唯一键值）
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
betType	Integer	交易记录类别： 1: 真实游戏
transactionType	Integer	交易类别： 1: 现金 2: 红利 3: 免费游戏
platform	Integer	投注记录平台 (请参阅平台以获取更多信息)
currency	String	记录中玩家所使用的货币
betAmount	Decimal*	玩家投注
winAmount	Decimal*	玩家派彩
jackpotContributionAmount	Decimal*	玩家的奖池贡献额
jackpotWinAmount	Decimal*	玩家的奖池所赢金额
balanceBefore	Decimal*	交易前的玩家余额
balanceAfter	Decimal*	交易后的玩家余额
handsStatus	Integer	投注状态： 1: 非最后一手投注 2: 最后一手投注 3: 已调整
rowVersion	Integer	数据的更新时间 (以毫秒为单位的 Unix 时间戳)
betTime	Integer	当前投注的开始时间 (以毫秒为单位的 Unix 时间戳)
betEndTime	Integer	当前投注的结束时间 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额，系统将可能返回多达 6 个小数位。

例子:

```
{
  "data": [
    {
      "betId": 35677059,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 3000,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88081189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546511407,
      "betTime": 1529546613715,
      "betEndTime": 1529546700260
    },
    {
      "betId": 35677061,
      "parentBetId": 35677059,
      "playerName": "player1",
      "currency": "CNY",
      "gameId": 19,
      "platform": 2,
      "betType": 1,
      "transactionType": 1,
      "betAmount": 0,
      "winAmount": 0,
      "jackpotContributionAmount": 0,
      "jackpotWinAmount": 0,
      "balanceBefore": 88078189.2,
      "balanceAfter": 88078189.2,
      "handsStatus": 1,
      "rowVersion": 1529546556399,
      "betTime": 1529546647867,
      "betEndTime": 1529546701360
    }
  ],
  "error": null
}
```

## 2.4.11 获取失败投注

检索失败的投注：

注意

- 运营商可获得最近 60 天的投注历史记录。

请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetFailedBets

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
from_time	Long	是	查询开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) 注： 值为 1 - 40 天
to_time	Long	是	
count	Integer	是	个别批次的记录数量 注： 值为 1500 -5000

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&count=5000&from\_time=1346592723000&to\_time=1546592723000



## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
betId	Integer	子投注的唯一标识符 (唯一键值)
playerName	String	玩家的唯一标识符
message	String	失败原因
betStartTime	Integer	当前投注的开始时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": [
    {
      "betId": 10832988,
      "playerName": "testplayer1",
      "message": "Bet not found.",
      "betStartTime": 1600165980624
    }, {
      "betId": 10832989,
      "playerName": " testplayer2",
      "message": "Bet not found.",
      "betStartTime": 1600165980624
    }
  ],
  "error": null
}
```

## 2.4.12 待处理投注

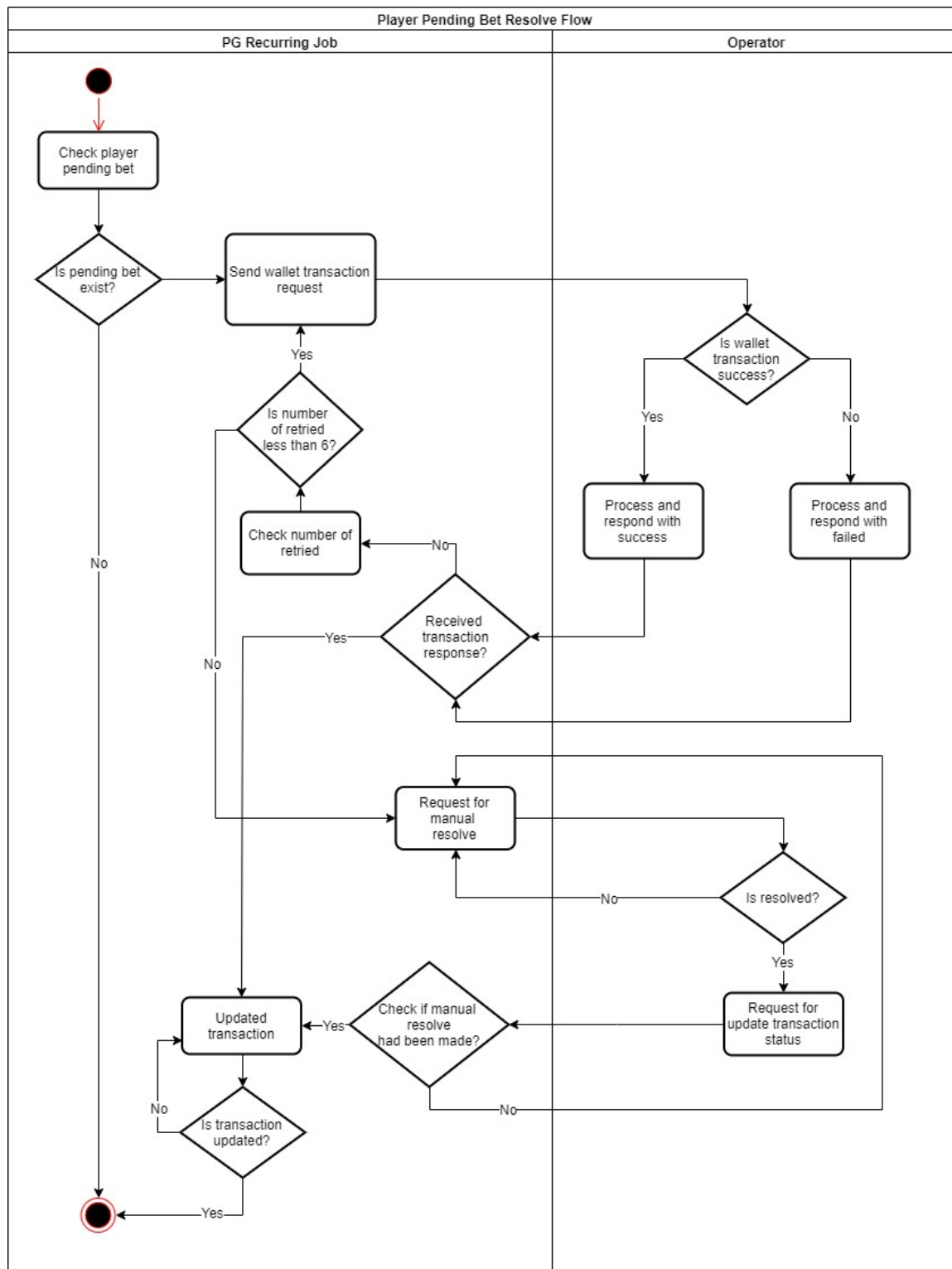
待处理投注将可能由于玩家游戏时所发生的不可预见的情况（例如：网络延迟，系统故障，系统超时等）而产生。

待处理投注通常会在玩家重新启动游戏时自动解决，但在某些罕见的情况下，该投注将保持在待处理状态：

- 遇到错误后，玩家未重新启动游戏。
- 解决待处理的投注时，PG Soft 系统未收到来自运营商的任何响应。
- 解决待处理的投注时，PG Soft 系统收到来自运营商的错误响应。

待处理投注解决时间的默认间隔为 60 分钟。超过该时间间隔的待处理投注将由 PG Soft 系统自动解决。此时间间隔是可自定义的，每个运营商都可自定义各自解决投注的时间间隔。

若在几次尝试之后，待处理投注仍解决失败，则需要由运营商手动解决该待处理投注。PG Soft 会在获得运营商的确认之后，将这些投注状态标记为完成。



### 2.4.12.1. 获取待处理投注

检索待处理投注：

注意

- 运营商可获得最近 60 天的投注历史记录。

请求

API URL : {DataGrabAPIDomain}/Bet/v4/GetPendingBets

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_names	string	是	玩家的唯一标识符
from_time	Long	是	查询开始和结束的时间范围 (以毫秒为单位的 Unix 时间戳) <b>注： 值为 1 - 40 天</b>
to_time	Long	是	
count	Integer	是	个别批次的记录数量 <b>注： 值为 1500 - 5000</b>

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_names=testplayer1&player\_names=testplayer2&count=5000&from\_time=1346592723000&to\_time=1546592723000

返回结果

参数:

JSON 数组

参数名称	数据类型	说明
betId	Integer	子投注的唯一标识符 (唯一键值)
playerName	String	玩家的唯一标识符
gameId	String	游戏的唯一标识符
betStatus	Integer	投注状态 1: 待处理

例子:

```
{
  "data": [
    {
      "playerName": "qa9",
      "gameId": 7,
      "betStatus": 1,
      "betId": 1325714346441543680
    }, {
      "playerName": "qa9",
      "gameId": 7,
      "betStatus": 1,
      "betId": 1325714346441543681
    }
  ],
  "error": null
}
```

## 2.4.12.2. 手动解决待处理投注

手动解决待处理投注：

请求

API URL : {DataGrabAPIDomain}/Bet/v4/ManualResolveBet

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	string	是	玩家的唯一标识符
bet_id	Integer	是	子投注的唯一标识符（唯一键值）
is_success_bet	Boolean	是	投注状态： True: 投注成功 False: 投注失败
game_id	Integer	是	游戏的唯一标识符
balance_before_adjustment	Decimal	否	调整前的玩家余额（仅适用于成功的投注）

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=testplayer1&bet\_id=1325714346441543680&is\_success\_bet=true&game\_id=1

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
resolveStatus	Boolean	投注的解决结果: True: 成功 False: 失败
betStatus	Integer	投注状态: 1: 待处理 2: 失败 3: 成功 4: 已调整

例子:

```
{
  "data": {
    "resolveStatus": true,
    "betStatus": 3
  },
  "error": null
}
```

### 2.4.13 手动重置待处理游戏状态

手动重置待处理游戏的状态：

请求

API URL : {DataGrabAPIDomain}/Bet/v4/ManualResetGameState

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	string	是	玩家的唯一标识符
game_id	Integer	是	游戏的唯一标识符
operator_player_session	string	否	玩家的运营商会话

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=testplayer1&game\_id=1

返回结果

参数：

JSON 对象

参数名称	数据类型	说明
resetStatus	Boolean	投注的解决结果： True: 成功 False: 失败

例子：

```
{
  "data": {
    "resetStatus": true
  },
  "error": null
}
```



## 2.5 游戏 [最新版本: v2]

### 2.5.1 获取游戏列表

获取游戏列表:

请求

API URL : {PgSoftAPIDomain}/Game/v2/Get

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	游戏合法投注金额的货币
language	String	否	数据内容的语言 en-us: 英文 zh-cn: 中文

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&currency=CNY&language=en-us

返回结果

参数:

JSON 数组

参数名称	数据类型	说明
gameId	Integer	游戏的唯一标识符
gameName	String	游戏名称
gameCode	String	游戏识别码
gameLegalBetAmounts	JSON Array	游戏的合法投注金额
status	Integer	全球游戏状态: 0: 无效 1: 活跃 2: 已暂停

		<b>注：唯有在游戏状态和发布状态都显示活跃时才能进入游戏</b>
releaseStatus	Integer	运营商的游戏状态： 0: 无效 1: 活跃 2: 已暂停 <b>注：唯有在游戏状态和发布状态都显示活跃时才能进入游戏</b>

#### gameLegalBetAmounts JSON 数组

参数名称	数据类型	说明
gameId	Integer	游戏的唯一标识符
gameTypeId	Integer	游戏类型（仅限卡牌游戏）： 0: 无 1: 百家乐 – 免佣 2: 百家乐 – 传统
legalBetAmounts	JSON Array	游戏里可用的合法投注金额

#### legalBetAmounts JSON 数组

参数名称	数据类型	说明
coinSize	Decimal	游戏的投注大小
betMultiplier	Integer	游戏的投注级别
baseBet	Integer	游戏的基本投注
betAmount	Decimal	游戏的投注金额

例子:

```
{
  "data": [
    {
      "gameId": 1,
      "gameName": "Honey Trap of Diao Chan",
      "gameCode": "diaochan",
      "status": 1,
      "releaseStatus": 1,
      "gameLegalBetAmounts": [{
        "gameId": 1,
        "gameTypeId": 0,
        "legalBetAmounts": [
          {
            "coinSize": 0.03,
            "betMultiplier": 1,
            "baseBet": 30,
            "betAmount": 0.9
          },
          {
            "coinSize": 0.03,
            "betMultiplier": 2,
            "baseBet": 30,
            "betAmount": 1.8
          }
        ]
      }
    ]
  },
  {
    "gameId": 2,
    "gameName": "Gem Saviour",
    "gameCode": "gem-saviour",
    "status": 1,
    "releaseStatus": 1,
    "gameLegalBetAmounts": [
      {
        "gameId": 1,
        "gameTypeId": 0,
        "legalBetAmounts": [{
          "coinSize": 0.03,
          "betMultiplier": 1,
          "baseBet": 30,
          "betAmount": 0.9
        }
        ],
        {
          "coinSize": 0.03,
          "betMultiplier": 2,
          "baseBet": 30,
          "betAmount": 1.8
        }
      ]
    ]
  }
],
  "error": null
}
```

## 2.5.2 获取游戏的合法投注金额

获取游戏的合法投注金额。

### 请求

API URL : {PgSoftAPIDomain}/Game/v2/GetGameLegalBetAmounts

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	游戏合法投注金额的货币
game_ids	Integer Array	是	游戏的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&currency=CNY&game\_ids=1&game\_ids=2

### 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
gameId	Integer	游戏的唯一标识符
gameTypeId	Integer	游戏类型（仅限卡牌游戏）： 0: 无 1: 百家乐 – 免佣 2: 百家乐 – 传统
legalBetAmounts	JSON Array	游戏里可用的合法投注金额

legalBetAmounts JSON 数组

参数名称	数据类型	说明
coinSize	Decimal	游戏的投注大小
betMultiplier	Integer	游戏的投注级别
baseBet	Integer	游戏的基本投注
betAmount	Decimal	游戏的投注金额

例子:

```
{
  "data": [
    {
      "gameId": 1,
      "gameTypeId": 0,
      "legalBetAmounts": [
        {
          "coinSize": 0.03,
          "betMultiplier": 1,
          "baseBet": 30,
          "betAmount": 0.9
        },
        {
          "coinSize": 0.03,
          "betMultiplier": 2,
          "baseBet": 30,
          "betAmount": 1.8
        }
      ]
    }
  ],
  "error": null
}
```

### 2.5.3 更改游戏发布状态

更改游戏发布状态：

请求

API URL : {PgSoftAPIDomain}/Game/v2/UpdateGameStatus

HTTP 方法 : POST

参数

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
release_status	Integer	是	游戏的发布状态： 0: 无效 1: 活跃
game_ids	Integer Array	是	游戏的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&release\_status=1&game\_ids=1&game\_ids=2

请求

参数：

JSON 对象

参数名称	数据类型	说明
actionResult	Boolean	操作请求的返回结果： true: 成功 false: 失败

例子：

```
{
  "data":{
    "actionResult": true
  },
  "error": null
}
```

## 2.6 累积奖池 [最新版本: v1]

### 2.6.1 获取累积奖池列表

获取累积奖池列表:

请求

API URL : {PgSoftAPIDomain}/Jackpot/v1/Get

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
currency	String	是	奖池货币
game_id	Integer	否	游戏的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&currency=cny

返回结果

参数:

JSON 数组

参数名称	数据类型	说明
jackpotName	String	奖池名称
jackpotRoomId	String	奖池房间 ID
amount	Decimal*	奖池总额
jackpotType	Integer	奖池类型: 1: 至尊奖 2: 大奖 3: 好运奖
poolId	Integer	奖池的唯一 ID
gameIds	Integer Array	游戏的唯一标识符
lastUpdatedOn	Long	奖池的更新日期 (以毫秒为单位的 Unix 时间戳)

\*根据游戏和余额, 系统将可能返回多达 6 个小数位数。

例子:

```
{
  "data": [
    {
      "jackpotName": "Operator1 Dragon Jackpot Grand",
      "jackpotRoomId": "JackpotRoom5",
      "amount": 50833.53,
      "jackpotType": 1,
      "poolId": 2,
      "gameIds": [
        9,
        23,
        46,
        47
      ],
      "lastUpdatedOn": 1551697413000
    },
    {
      "jackpotName": "Operator1 Dragon Jackpot Major",
      "jackpotRoomId": "JackpotRoom6",
      "amount": 9387.1,
      "jackpotType": 2,
      "poolId": 2,
      "gameIds": [
        9,
        23,
        46,
        47
      ],
      "lastUpdatedOn": 1551697413000
    }
  ],
  "error": null
}
```



## 2.7 玩家 [最新版本: v3]

### 2.7.1 获取在线玩家人数

获取在线的玩家人数：

请求

API URL : {PgSoftAPIDomain}/Player/v2/GetOnlinePlayerCount

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
game_id	Integer	否	游戏的唯一标识符 <b>注意：若未提供参数值，系统将返回所有游戏的在线玩家人数。</b>

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&game\_id=1

返回

参数

JSON 对象

参数名	数据类型	说明
totalCount	Integer	记录总数
totalPlayerCount	Integer	在线玩家总数
onlinePlayerSummary	JSON Array	游戏的在线玩家列表

JSON 数组

参数名	数据类型	说明
rowVersion	Long	数据的更新时间 (以毫秒为单位的 Unix 时间戳)
gameId	Integer	游戏的唯一标识符
playerCount	Integer	在线玩家总数
status	Integer	玩家状态 1: 在线

例子:

```
{
  "data": {
    "onlinePlayerSummary": [
      {
        "rowVersion": 1581073260000,
        "gameId": 0,
        "playerCount": 1,
        "status": 1
      },
      {
        "rowVersion": 1581073260000,
        "gameId": 52,
        "playerCount": 1,
        "status": 1
      },
      {
        "rowVersion": 1581073260000,
        "gameId": 68,
        "playerCount": 1,
        "status": 1
      }
    ],
    "totalCount": 3,
    "totalPlayerCount": 3
  },
  "error": null
}
```

## 2.7.2 获取玩家的线上状态

获取最后 10 名玩家的线上状态：

请求

API URL : {PgSoftAPIDomain}/Player/v2/GetPlayersOnlineStatus

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_names	String Array	是	玩家的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_names=player1&player\_names=player2

返回

参数

JSON 数组

参数名称	数据类型	说明
playerName	String	玩家的唯一标识符
gameId	String	游戏的唯一标识符
status	Integer	玩家的线上状态 0: 离线 1: 在线

例子：

```
{
  "data": [
    {
      "playerName": "player1",
      "status": 1,
      "gameId": 12
    },
    {
      "playerName": "player2",
      "status": 0
    }
  ],
  "error": null
}
```

### 2.7.3 获取在线玩家列表

获取在线玩家列表

请求

API URL : {PgSoftAPIDomain}/Player/v3/GetOnlinePlayers

HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
row_count	Integer	否	个别批次的记录数量 <b>注意:</b> <ul style="list-style-type: none"> <li>默认值: 1</li> <li>值度: 1-5000</li> </ul>

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&row\_count=10

## 返回结果

参数:

JSON 数组

参数名称	数据类型	说明
playerName	String	玩家的唯一标识符
gameId	Integer	游戏的唯一标识符
rowVersion	Integer	数据的更新时间 (以毫秒为单位的 Unix 时间戳)

例子:

```
{
  "data": [
    {
      "playerName": "tesplayer01",
      "gameId": 60,
      "rowVersion": 1607503788000
    },
    {
      "playerName": "tesplayer02",
      "gameId": 65,
      "rowVersion": 1607503809000
    }
  ],
  "error": null
}
```

## 2.7.4 创建玩家

在 PG 游戏系统中创建新玩家帐户：

### 注意

由于玩家帐户将被自动创建，以供初次访问 PG 游戏的玩家使用，所以此 API 的执行是非强制性的。

### 请求

API URL : {PgSoftAPIDomain}/v3/Player/Create

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符 <ul style="list-style-type: none"> <li>• 玩家账号中的字母不区分大小写</li> <li>• 只允许使用字母、数字和 "@"、"-","_" 符号</li> </ul> <b>注：最多 50 个字符</b>
nickname	String	否	玩家的用户名 <b>注：最多 50 个字符</b>
currency	String	是	玩家货币

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player123&currency=EUR&nickname=player123

## 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
action_result	Integer	请求状态 1: 成功 0: 失败

例子:

```
{
  "data": {
    "action_result": 1
  },
  "error": null
}
```

## 错误的返回结果

代码	信息
1034	请求无效
1035	操作失败
1200	内部服务器错误
1204	无效的运营商
1305	无效的玩家 (玩家已存在)
1315	玩家正在进行另一项操作

## 2.7.5 踢出玩家

将玩家从 PG 游戏中踢出，玩家仍然可以重新登录游戏：

请求

API URL : {PgSoftAPIDomain}/v3/Player/Kick  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player123

### 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
action_result	Integer	请求状态 1: 成功 0: 失败

例子：

```
{
  "data": {
    "action_result": 1
  },
  "error": null
}
```

### 错误的返回结果

代码	信息
1034	请求无效
1035	操作失败
1200	内部服务器错误
1204	无效的运营商
1305	无效的玩家（玩家已存在）
1315	玩家正在进行另一项操作



## 2.7.6 禁止玩家

禁止玩家帐户，被禁止的玩家将无法再访问 PG 游戏。

### 请求

API URL : {PgSoftAPIDomain}/v3/Player/Suspend  
HTTP 方法 : POST

参数:

参数名称	数据类型	强制与否?	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符

例子:

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player123

### 返回结果

参数:

JSON 对象

参数名称	数据类型	说明
action_result	Integer	请求状态 1: 成功 0: 失败

例子:

```
{
  "data": {
    "action_result": 1
  },
  "error": null
}
```

### 错误的返回结果

代码	信息
1034	请求无效
1035	操作失败
1200	内部服务器错误
1204	无效的运营商
1305	无效的玩家 (玩家已存在)
1315	玩家正在进行另一项操作

## 2.7.7 恢复玩家

恢复被禁止的玩家帐户：

请求

API URL : {PgSoftAPIDomain}/v3/Player/Reinstate

HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player123

返回结果

参数：

JSON 对象

参数名称	数据类型	说明
action_result	Integer	请求状态 1: 成功 0: 失败

例子：

```
{
  "data": {
    "action_result": 1
  },
  "error": null
}
```

错误的返回结果

代码	信息
1034	请求无效
1035	操作失败
1200	内部服务器错误
1204	无效的运营商
1305	无效的玩家（玩家已存在）
1315	玩家正在进行另一项操作

## 2.7.8 检查玩家状态

查看特定玩家的帐户状态：

### 注意

此 API 不用于检查在线玩家的状态，而仅用于检查玩家账户在 PG 系统中的状态。请使用后台以查看线上活跃的玩家。

### 请求

API URL : {PgSoftAPIDomain}/v3/Player/Check  
HTTP 方法 : POST

参数：

参数名称	数据类型	强制与否？	说明
operator_token	String	是	运营商的唯一标识符
secret_key	String	是	PGSoft 和运营商之间的共享密码
player_name	String	是	玩家的唯一标识符

例子：

operator\_token=abc123&secret\_key=a1b25cde5f3gh46ijkl&player\_name=player123

### 返回结果

参数：

JSON 对象

参数名称	数据类型	说明
player_name	String	玩家的唯一标识符
status	Integer	请求状态 0: 已关闭 1: 活跃 2: 不活跃/已冻结 3: 已禁止 4: 玩家正在进行另一项操作

例子：

```
{
  "data": {
    "player_name": "player123",
    "status": 1
  },
  "error": null
}
```

## 3. 附录

### 3.1 平台

代码	类别	平台
1	Web	Windows
2		macOS
3		Android
4		iOS
5		其它
6	Cordova	Android
7		iOS
8	Electron	Windows
9		macOS
10	Native	Windows
11		macOS
12		Android
13		iOS
98	others	System
99		挂机系统