PG SOFT

# Integration Document

External API v1.11

POCKET GAMES SOFT ™

# Contents

Contents ... 1

Revision History ... 4

1. Overview ... 11

2. API Methods ... 12

- 2.1 API Formats ... 12
  - 2.1.1 Request ... 12
  - 2.1.2 Response ... 12
  - 2.1.3 Hash Authentication (Optional) ... 13
- 2.2 Free Game [Latest version: v1] ... 16
  - 2.2.1 Get Free Games List ... 16
  - 2.2.2 Get List of Free Game Converted to Bonus Game/Cash ... 21
  - 2.2.3 Get List of Free Games that have Converted to Bonus Game then Cash ... 23
  - 2.2.4 Get Free Game Players List ... 26
  - 2.2.5 Get List of Unregistered Players in Free Game ... 30
  - 2.2.6 Get All Player Details for Free Game ... 33
  - 2.2.7 Get Single Player Details in Free Game ... 37
  - 2.2.8 Create Free Game ... 41
  - 2.2.9 Create Free Game by Bet Amount ... 44
  - 2.2.10 Transfer Player to Standard Free Game ... 47
  - 2.2.11 Transfer Player to Flexible Free Game ... 49
  - 2.2.12 Cancel Free Game ... 52
  - 2.2.13 Cancel Free Game of a Player ... 53
  - 2.2.14 Cancel Free Game of an Unregistered Player ... 55
  - 2.2.15 Cancel All Free Games of a Player ... 56
  - 2.2.16 Get Free Game Player Summary ... 57
- 2.3 Bonus [Latest version: v1] ... 60
  - 2.3.1 Get Bonus Game List ... 60
  - 2.3.2 Get List of Bonus Games Converted to Cash ... 64
  - 2.3.3 Get Bonus Game Players List ... 67
  - 2.3.4 Get List of Unregistered Players of Bonus Game ... 70
  - 2.3.5 Get All Player Details in Bonus Game ... 73
  - 2.3.6 Get Single Player Details in Bonus Game ... 76
  - 2.3.7 Create Bonus Game ... 79

2.3.8 Transfer Player to Standard Bonus Game ........ 82
2.3.9 Transfer Player to Flexible Bonus Game ........ 84
2.3.10 Cancel Bonus Game ........ 87
2.3.11 Cancel Bonus Games of Registered Player ........ 88
2.3.12 Cancel Bonus Game of Unregistered Player ........ 89
2.3.13 Cancel All Bonus Games of a Player ........ 90
2.4 Tournament [Latest version: v2] ........ 91
2.4.1 Get Tournaments List ........ 91
2.4.2 Get Tournament Players List ........ 96
2.4.3 Get Tournament Top Rankings ........ 99
2.4.4 Register Tournament Players ........ 101
2.4.5 Get Tournament Cash Prize Reward List ........ 104
2.4.6 Create Tournament ........ 107
2.4.7 Update Tournament ........ 110
2.4.8 Update Sub-Tournament Information ........ 112
2.4.9 Get Tournament Information ........ 114
2.4.10 Create Tournament Information ........ 116
2.4.11 Update Tournament Information ........ 118
2.4.12 Delete Tournament Information ........ 120
2.4.13 Remove a Tournament Player ........ 121
2.4.14 Revive Players ........ 122
2.5 Bet History [Latest version: v4] ........ 124
2.5.1 Get History ........ 125
2.5.2 Get History for Specific Time ........ 129
2.5.3 Get Single Player History ........ 133
2.5.4 Get Player’s Daily Summary ........ 136
2.5.5 Get Player’s Daily Summary for Specific Time ........ 139
2.5.6 Get Hands Summary ........ 142
2.5.7 Get Hourly Hands Summary ........ 145
2.5.8 Get Summary of Total Player Bets ........ 148
2.5.9 Get Player’s Unfinished Games ........ 152
2.5.10 Get Bet Details ........ 155
2.5.11 Get Failed Bets ........ 158
2.5.12 Pending Bets ........ 160

2.5.13 Get Pending Bets ..... 162
2.5.14 Manual Resolve of Pending Bet ..... 164
2.5.15 Manual Reset of Pending Game State ..... 166
2.6 Game [Latest version: v2] ..... 168
2.6.1 Get Games List ..... 168
2.6.2 Get Game Legal Bet Amounts ..... 171
2.6.3 Change Operator’s Game Release Status ..... 173
2.6.4 Get Game Cert ..... 175
2.7 Cash [Latest version: v2] ..... 177
2.7.1 Get Wallet Transaction ..... 177
2.7.2 Get Manual Adjustment Details ..... 179
2.8 Player [Latest version: v3] ..... 182
2.8.1 Get Online Player Count ..... 182
2.8.2 Get Players’ Online Status ..... 184
2.8.3 Get Online Player List ..... 186
2.8.4 Create Player ..... 188
2.8.5 Kick Out Player ..... 190
2.8.6 Suspend Player ..... 192
2.8.7 Resume Player ..... 194
2.8.8 Check Player Status ..... 196
3. Appendix ..... 198
3.1.1 Platform ..... 198

# Revision History

| Version | Date | Description |
|---|---|---|
| 1.0 | 2019-04-03 | First draft |
| 1.1 | 2019-04-12 | Added new content<br>• 2.1.5 Get Free Game Unregistered Players<br>• 2.1.12 Cancel All Free Games of Player<br>• 2.2.4 Get Bonus Unregistered Players<br>• 2.2.11 Cancel All Bonus of Player<br>• 2.4.6 Get Hands Summary<br>Amended Overview<br>Amended API request parameter<br>• 2.1.6 Create Free Game<br>• 2.1.11 Cancel Unregistered Player Free Game<br>• 2.2.5 Create Bonus<br>• 2.2.10 Cancel Unregistered Player Bonus<br>Amended API description<br>• 2 API Methods<br>• 2.1.2 Get List of Free Game Converted to Bonus<br>• 2.1.3 Get List of Free Game's Bonus Converted to Cash<br>• 2.1.10 Cancel Registered Player Free Game<br>• 2.1.11 Cancel Unregistered Player Free Game<br>• 2.2.2 Get List of Bonus Converted to Cash<br>• 2.2.9 Cancel Registered Player Bonus<br>• 2.2.10 Cancel Unregistered Player Bonus |
| 1.2 | 2019-04-15 | Added notes for all bet history APIs (2.4) |
| 1.3 | 2019-04-17 | Added new contents<br>• 2.1.6 Get Free Game All Players<br>• 2.1.7 Get Player Free Games<br>• 2.2.5 Get Bonus All Players<br>• 2.2.6 Get Player Bonus<br>Amended API descriptions & request descriptions<br>• 2.1.4 Get Free Game Players<br>• 2.1.5 Get Free Game Unregistered Players<br>• 2.2.3 Get Bonus Players<br>• 2.1.5 Get Bonus Unregistered Players<br>• 2.4.6 Get Hands Summary<br>• 2.5.1 Get Game List<br>Amended API request<br>• 2.1.5 Get Free Game Unregistered Players |

| | | |
|---|---|---|
| | | • 2.1.7 Create Free Game<br>• 2.2.4 Get Bonus Unregistered Players<br>• 2.2.6 Create Bonus<br>Amended API response<br>• 2.1.6 Get Free Game All Players<br>• 2.1.7 Create Free Game<br>• 2.2.5 Get Bonus All Players<br>• 2.2.6 Create Bonus<br>• 2.4.6 Get Hands Summary<br>• 2.5.2 Get Game List |
| 1.4 | 2019-07-12 | Added new contents<br>• 2.1.9 Create Free Game by Bet Amount<br>• 2.3.1 Get Tournaments<br>• 2.3.2 Get Tournament Players<br>• 2.5.2 Get Game Legal Bet Amount<br>Amended API<br>• 2.5.1 Get Game List (version 2)<br>Amended API request<br>• 2.1.8 Create Free Game<br>Amended API response<br>• 2.1.1 Get Free Games |
| 1.5 | 2019-08-22 | Amended API description<br>• 2.1.1 Get Free Games<br>• 2.1.2 Get List of Free Game Converted to Bonus/Cash<br>• 2.1.3 Get List of Free Game’s Bonus Converted to Cash<br>• 2.1.4 Get Free Game Players<br>• 2.1.6 Get Free Game All Players<br>• 2.2.1 Get Bonus<br>• 2.2.2 Get List of Bonus Converted to Cash<br>• 2.2.3 Get Bonus Players<br>• 2.2.5 Get Bonus All Players<br>• 2.3.1 Get Tournaments<br>• 2.3.2 Get Tournament Players<br>• 2.4.5 Get Player Daily Summary for Specific Time<br>Amend API request<br>• 2.1.7 Get Player Free Games<br>• 2.2.9 Transfer Player to Flexible Bonus<br>• 2.4.1 Get History<br>Amended API response<br>• 2.4.1 Get History<br>• 2.4.2 Get History for Specific Time |

| | | |
|---|---|---|
| | | • 2.4.3 Get Single Player History<br>• 2.4.6 Get Hands Summary<br>Added new contents<br>• 2.4.7 Get Player Bets Summation Total |
| 1.6 | 2019-11-13 | Added new contents<br>• 2.4.7 Get Hourly Hands Summary<br>Amended API version<br>• 2.4 Bet History<br>Amended API description<br>• 2.4 Bet History<br>• 2.4.1 Get History<br>Amended API response<br>• 2.4.1 Get History<br>• 2.4.2 Get History for Specific Time<br>• 2.4.3 Get Single Player History |
| 1.7 | 2020-02-10 | Added new contents<br>• 2.5.3 Change Game Release Status<br>• 2.7.1 Get Online Player Count<br>• 2.7.2 Get Player Online Status<br>Amended API response<br>• 2.4.1 Get History<br>• 2.4.2 Get History for Specific Time<br>• 2.4.3 Get Single Player History<br>• 2.4.4 Get Player Daily Summary<br>• 2.4.5 Get Player Daily Summary for Specific Time<br>• 2.4.6 Get Hands Summary<br>• 2.4.7 Get Hourly Hands Summary<br>• 2.4.8 Get Player Bets Summation Total<br>• 2.6.1 Get Jackpot List<br>Amended API response parameter description<br>• 2.1.6 Get Free Game All Players<br>• 2.1.7 Get Player Free Games<br>• 2.2.5 Get All Bonus Players<br>• 2.2.6 Get Player Bonus |
| 1.8 | 2020-12-11 | Added new API version<br>• 2.7 Player<br>Added new API<br>• 2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash<br>• 2.4.13 Pending Bets<br>• 2.4.14 Manual Reset Pending Game State<br>• 2.7.3 Get Online Players |

| | | |
|---|---|---|
| | | • 2.7.4 Create Player<br>• 2.7.5 Kick Out Player<br>• 2.7.6 Suspend Player<br>• 2.7.7 Resume Player<br>• 2.7.8 Check Player Status<br>Amended API title or description<br>• 2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash<br>• 2.1.10 Transfer Player to Standard Free Game<br>• 2.1.13 Cancel Player Free Game<br>• 2.2.6 Get Single Player Details in Bonus Game<br>• 2.2.7 Create Bonus Game<br>• 2.4 Bet History<br>• 2.4.1 Get History<br>• 2.4.2 Get History for Specific Time<br>• 2.4.3 Get Single Player History<br>• 2.4.4 Get Player's Daily Summary<br>• 2.4.5 Get Player's Daily Summary for Specific Time<br>• 2.4.6 Get Hands Summary<br>• 2.4.7 Get Hourly Hands Summary<br>• 2.4.8 Get Summary of Total Player Bets<br>• 2.4.9 Get Player's Unfinished Games<br>• 2.4.10 Get All Unsuccessful Bets<br>• 2.4.11 Get Bet Details<br>• 2.4.12 Get Failed Bets<br>• 2.6.1 Get Jackpots List<br>• 2.7.2 Get Players Online Status<br>Amended API request or response contents<br>• 2.1.1 Get Free Games List<br>• 2.1.4 Get Free Game Players List<br>• 2.1.5 Get List of Unregistered Players in Free Game<br>• 2.1.6 Get All Player Details for Free Game<br>• 2.1.7 Get Single Player Details in Free Game<br>• 2.1.8 Create Free Game<br>• 2.1.9 Create Free Game by Bet Amount<br>• 2.1.10 Transfer Player to Standard Free Game<br>• 2.1.11 Transfer Player to Flexible Free Game<br>• 2.2.1 Get Bonus Game List<br>• 2.2.2 Get List of Bonus Games Converted to Cash<br>• 2.2.3 Get Bonus Game Players List<br>• 2.2.4 Get List of Unregistered Players of Bonus Game<br>• 2.2.5 Get All Player Details in Bonus Game<br>• 2.2.6 Get Single Player Details in Bonus Game<br>• 2.2.9 Transfer Player to Flexible Bonus Game<br>• 2.2.13 Cancel All Bonus Games of a Player<br>• 2.3.1 Get Tournaments List |

| | | |
|---|---|---|
| | | • 2.3.2 Get Tournament Players List<br>• 2.3.4 Register Tournament Players<br>• 2.3.5 Get Tournament Cash Prize Reward List<br>• 2.4.1 Get History<br>• 2.4.2 Get History for Specific Time<br>• 2.4.3 Get Single Player History<br>• 2.4.4 Get Player’s Daily Summary<br>• 2.4.5 Get Player’s Daily Summary for Specific Time<br>• 2.4.6 Get Hands Summary<br>• 2.4.7 Get Hourly Hands Summary<br>• 2.4.8 Get Summary of Total Player Bets<br>• 2.4.9 Get Player’s Unfinished Games<br>• 2.4.10 Get All Unsuccessful Bets<br>• 2.4.11 Get Bet Details<br>• 2.4.12 Get Failed Bets<br>• 2.6.1 Get Jackpots List<br>• 2.7.2 Get Players Online Status |
| 1.9 | 2021-05-20 | Added new API request parameter<br>• All APIs in 2. API Methods<br>Added new APIs<br>• 2.4.6 Create Tournament<br>• 2.4.7 Update Tournament<br>• 2.4.8 Update Sub-Tournament<br>• 2.4.9 Get Tournament Info Language By Tournament Id<br>• 2.4.10 Create Tournament Info Language<br>• 2.4.11 Update Tournament Info Language<br>• 2.4.12 Delete Tournament Info Language<br>• 2.4.13 Cancel Tournament Player<br>• 2.4.14 Revive Players<br>Amend API response<br>• 2.5.7 Get Hourly Hands Summary<br>Amended API response description<br>• 2.5.7 Get Hourly Hands Summary<br>• 2.5.8 Get Summary of Total Player Bets<br>Amend API request<br>• 2.5.8 Get Summary of Total Player Bets<br>• 2.5.9 Get Player’s Unfinished Games<br>Added new contents<br>• 2.1.3 Hash Authentication |
| **1.10** | 2021-10-13 | Amended API request or response contents<br>• 2.1.3 Hash Authentication<br>• 2.2.4 Get Free Game Players List<br>• 2.2.6 Get All Player Details for Free Game<br>• 2.2.7 Get Single Player Details in Free Game<br>• 2.2.8 Create Free Game |

| | | |
|---|---|---|
| | | • 2.2.9 Create Free Game by Bet Amount<br>• 2.2.10 Transfer Player to Standard Free Game<br>• 2.2.11 Transfer Player to Flexible Free Game<br>• 2.2.15 Cancel All Free Games of a Player<br>• 2.3.3 Get Bonus Game Players List<br>• 2.3.5 Get All Player Details in Bonus Game<br>• 2.3.6 Get Single Player Details in Bonus Game<br>• 2.3.7 Create Bonus Game<br>• 2.3.8 Transfer Player to Standard Bonus Game<br>• 2.3.9 Transfer Player to Flexible Bonus Game<br>• 2.4.1 Get Tournaments List<br>• 2.4.5 Get Tournament Cash Prize Reward List<br>• 2.4.8 Update Sub-Tournament Information<br>• 2.4.10 Create Tournament Information<br>• 2.4.11 Update Tournament Information<br>• 2.5.1 Get History<br>• 2.5.2 Get History for Specific Time<br>• 2.5.3 Get Single Player History<br>• 2.5.6 Get Hands Summary<br>• 2.5.7 Get Hourly Hands Summary<br>• 2.5.8 Get Summary of Total Player Bets<br>• 2.5.9 Get Player’s Unfinished Games<br>• 2.5.10 Get Bet Details<br>• 2.5.13 Get Pending Bets<br>• 2.6.1 Get Games List<br>• 2.8.3 Get Online Player List<br>Added new API<br>• 2.2.16 Get Free Game Player Summary<br>• 2.6.4 Get Game Cert<br>• 2.7.1 Get Wallet Transaction<br>• 2.7.2 Get Manual Adjustment Details |
| **1.11** | 2022-06-27 | Removed API<br>• 2.8 Jackpot<br>Amended API request or response contents<br>• 2.1.3 Hash Authentication (Optional)<br>• 2.2.1 Get Free Games List<br>• 2.2.2 Get List of Free Game Converted to Bonus Game/Cash<br>• 2.2.3 Get List of Free Games that have Converted to Bonus Game then Cash<br>• 2.2.4 Get Free Game Players List |

- 2.2.5 Get List of Unregistered Players in Free Game
- 2.2.6 Get All Player Details for Free Game
- 2.2.7 Get Single Player Details in Free Game
- 2.2.8 Create Free Game
- 2.2.9 Create Free Game by Bet Amount
- 2.2.12 Cancel Free Game
- 2.2.13 Cancel Free Game of a Player
- 2.2.14 Cancel Free Game of an Unregistered Player
- 2.2.15 Cancel All Free Games of a Player
- 2.2.16 Get Free Game Player Summary
- 2.3.1 Get Bonus Game List
- 2.3.2 Get List of Bonus Games Converted to Cash
- 2.3.3 Get Bonus Game Players List
- 2.3.4 Get List of Unregistered Players of Bonus Game
- 2.3.5 Get All Player Details in Bonus Game
- 2.3.6 Get Single Player Details in Bonus Game
- 2.3.7 Create Bonus Game
- 2.3.10 Cancel Bonus Game
- 2.3.11 Cancel Bonus Games of Registered Player
- 2.3.12 Cancel Bonus Game of Unregistered Player
- 2.3.13 Cancel All Bonus Games of a Player
- 2.4.2 Get Tournament Players List
- 2.4.3 Get Tournament Top Rankings
- 2.4.4 Register Tournament Players
- 2.4.6 Create Tournament
- 2.5.1 Get History
- 2.5.2 Get History for Specific Time
- 2.5.3 Get Single Player History
- 2.5.5 Get Player’s Daily Summary for Specific Time
- 2.5.6 Get Hands Summary
- 2.5.8 Get Summary of Total Player Bets
- 2.5.10 Get Bet Details
- 2.5.13 Get Pending Bets
- 2.5.14 Manual Resolve of Pending Bet
- 2.5.15 Manual Reset of Pending Game State
- 2.6.1 Get Games List
- 2.7.1 Get Wallet Transaction
- 2.7.2 Get Manual Adjustment Details
- 2.8.3 Get Online Player List

# 1. Overview

The External API allows you to programmatically access PGSoft's tools and services. You can use these APIs to retrieve information, create, modify, or cancel various PG services such as free games, bonus games, tournament and more.

PGSoft APIs are divided by services. Each of these services has its own section in this documentation. All APIs in this documentation are optional, operator may choose to integrate API based on requirements.

# 2. API Methods

## 2.1 API Formats

### 2.1.1 Request

The external API uses HTTP form methods and a RESTful endpoint structure. You format requests in the following content type:
`Content-Type: application/x-www-form-urlencoded`

### 2.1.2 Response

API returns JSON-formatted responses. For successful and failed API request, operator is required to return an API response with HTTP status code 200.

API uses global JSON response format for all API methods,  with the following content type:
`Content-Type: application/json`

Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| **data**[1] | JSON Object | Response information for API methods |
| **error**[2] | JSON Object | Error information when an error or exception occur |

Example:

**Successful Response**

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

**Failed/Error Response**

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

---

[1] This field will show *null* value in error response

[2] This field will show *null* value in success API response. Please refer to error response section for error response format

### 2.1.3 Hash Authentication (Optional)

We highly recommend that the operator integrate hash authentication to secure every request. If the operator chooses to integrate with hash authentication, PG system will validate the hash information provided in the header fields for every request.

#### 2.1.3.1 Header Fields

The following table describes the various request headers in the preceding example:

| Component | Description |
|---|---|
| Host | API domain<br>**Example**:<br>`apiexample.pgsoft.com` |
| x-date | Current date (UTC) in YYYYMMDD format<br>**Example:**<br>`20190902` |
| x-content-sha256 | SHA256 hash of the request string body<br>**Example**:<br>`1700116101f424b9f6fc695b4dbaf2b7b0ee763ba1b3b53298e3069143ed46f1` |
| Authorization | Combination of various components for request authentication<br>**Example**:<br>`PWS-HMAC-SHA256`<br>`Credential=20190902/OPERATORTOKENEXAMPLE/pws/v1,SignedHeaders=host;x-content-sha256;x- date,Signature=d78220cf06ae85f9d1db11dad9c3fd926799619eab3d28574aadb8cf328cd7aa` |

**Example**

```
Host:
apiexample.pgsoft.com
x-date: 20190902
x-content-sha256:
1700116101f424b9f6fc695b4dbaf2b7b0ee763ba1b3b53298e3069143ed46f
1
Authorization: PWS-HMAC-SHA256
Credential=20190902/OPERATORTOKENEXAMPLE/pws/v1,SignedHeaders=
host;x-content-sha256;x-date,Signature=
d78220cf06ae85f9d1db11dad9c3fd926799619eab3d28574aadb8cf328cd7aa
```

## Header: x-content-sha256

Calculate a SHA256 hash of the request string body:

**Example**

POST body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&row_version=1346592723000
```

x-content-sha256:

```
1700116101f424b9f6fc695b4dbaf2b7b0ee763ba1b3b53298e3069143ed46f1
```

## Header: Authentication

The following table describes the various components of the Authorization header value in the preceding example:

| Component | Description |
|---|---|
| Credential | Your access key ID and the scope information which includes the date, operator token and service that were used to calculate the signature.<br>This string has the following form:<br>`{x-date}/{operator_token}/pws/v1`<br>**Example**:<br>`20190902/OPERATORTOKENEXAMPLE/pws/v1` |
| SignedHeaders | A semicolon-separated list of request headers that you will use to compute the Signature. The list includes header names only, and the header names must be in lowercase.<br>**Fixed Value:**<br>`host;x-content-sha256;x-date` |
| Signature | The 256-bit signature expressed as 64 lowercase hexadecimal characters.<br>**Example**:<br>`d78220cf06ae85f9d1db11dad9c3fd926799619eab3d28574aadb8cf328cd7aa` |

To calculate a signature, you first need a string that includes *{host}{x-content-sha256}{x-date}* to sign. You will then calculate a HMAC-SHA256 hash of the string to sign by using a signing key:

```
hmac-sha256({salt}, {host}{x-content-sha256}{x-date})
```

Calculate a HMAC-SHA256 hash of the string body:

**Example**

Salt : `SALTEXAMPLE`
Host : `apiexample.pgsoft.com`
x-date : `20190902`
x-content-sha256 :
`1700116101f424b9f6fc695b4dbaf2b7b0ee763ba1b3b53298e3069143ed46f1`

**Signature:**
`d78220cf06ae85f9d1db11dad9c3fd926799619eab3d28574aadb8cf328cd7aa`

## 2.2 Free Game [Latest version: v1]

### 2.2.1 Get Free Games List

To get the list of created free games.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGames
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| from_time | Long | Yes | Time range of free games' start times (Unix time stamp in milliseconds) |
| to_time | Long | Yes | Time range of free games' end times (Unix time stamp in milliseconds) |
| from_created_time | Long | No | Time range of free games' created times (Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| free_game_id | Integer | No | Unique identity of free game |
| currency | String | No | Currency of free game |
| free_game_name | String | No | Given name of free game |
| free_game_transfer_type | Integer | No | Transfer type for free game<br>0: Standard<br>3: Flexible |
| conversion_type | Char | No | Type of conversion after free game is completed:<br>C: Cash<br>B: Bonus |

| | | | |
|---|---|---|---|
| status | Integer | No | Status of free game:<br>0: Inactive<br>1: Active<br>2: Expired |

Example:

URL:
```
https://api.pg-bo.me/external/FreeGame/v1/GetFreeGames?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
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
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of free games |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorToken | String | Unique identity of operator |
| freeGameId | Integer | Unique identity of free game |
| freeGameName | String | Given name of free game |
| transaction_id | String | Unique identity of transaction |
| isEvent | Boolean | Type of free game:<br>True: Flexible<br>False: Standard |
| gameIds | Integer Array | Unique identity of games |
| totalGame | Integer | Total number of free games that will be given to a single player |
| currencyCode | String | Currency of free game |
| coinSize | Decimal | Coin size of free game |
| multiplier | Integer | Bet multiplier of free game |
| minimumConversionAmount | Decimal | The minimum cash conversion amount of free game |
| maximumConversionAmount | Decimal | The maximum cash conversion amount of free game |
| conversionType | Char | Type of conversion after free game is completed:<br>C: Cash |

| | | |
|---|---|---|
| | | B: Bonus |
| bonusRatio | Decimal | Wagering requirement for bonus game.<br>Refers to the multiple of the turnover in free game that needs to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game.<br>Refers to the total amount that needs to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonusMinimumConversionAmount | Decimal | The minimum cash conversion amount of bonus game |
| bonusMaximumConversionAmount | Decimal | The maximum cash conversion amount of bonus game |
| bonusType | Integer | Bonus game conversion type:<br>0: Bonus<br>1: Cash<br>2: Bonus & Cash |
| status | Integer | Status of free game:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired |
| freeGameCountGiven | Integer | Total number of given free games |
| numberOfPlayers | Integer | Total number of players in free game |
| expiredDate | Long | Expiry date of free game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Created date of free game<br>(Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who created the free game |
| updatedBy | String | API or BackOffice user who updated the free game |
| isSupressDiscard | Boolean | Permission to allow players to cancel offer<br>True: Do not allow player to cancel offer<br>False: Allows player to cancel offer |

Example:

```
{
    "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
    }
```

### 2.2.2 Get List of Free Game Converted to Bonus Game/Cash

To get list of free games that has been converted to bonus game or cash.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameConvertedWallet
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| free_game_id | Integer | Yes | Unique identity of free game |
| from_transaction_time | Long | Yes | Time range of free game player's transactions<br>(Unix time stamp in milliseconds) |
| to_transaction_time | Long | Yes | |
| player_name | String | No | Unique identity of player<br>**Note: Fuzzy matching is implemented** |

Example:

URL:

```
https://api.pg-
bo.me/external/FreeGame/v1/GetFreeGameConvertedWallet?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&f
ree_game_id=12345&player_name=player1&from_transaction_time=1346592723000&to_trans
action_time=1346592723000
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalConvertedAmount | Decimal | Total amount that has been converted to cash in free game |
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of free games |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player’s free game wallet |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| gameId | Integer | Unique identity of game |
| transactionAmount | Decimal | Converted amount of player |
| transactionDateTime | Long | Time of conversion |

Example:

```
{
    "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 Get List of Free Games that have Converted to Bonus Game then Cash

To get the list of games that were converted from a free game to a bonus game, and then from bonus game to cash.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameBonusConvertedWallet
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| free_game_id | Integer | Yes | Unique identity of free game |
| from_transaction_time | Long | Yes | Time range of free game player's transactions<br>(Unix time stamp in milliseconds) |
| to_transaction_time | Long | Yes | |
| player_name | String | No | Unique identity of player<br>**Note: Fuzzy matching is implemented** |
| status | Integer | No | Status of bonus game:<br>0: Inactive<br>1: Active<br>2: Expired |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/GetFreeGameBonusConvertedWallet?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
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
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalConvertedAmount | Decimal | Total amount that has been converted to cash in free game |
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of free game players' details |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player's free game wallet |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| gameId | Integer | Unique identity of game |
| transactionAmount | Decimal | Amount converted of player |
| transactionDateTime | Long | Time of convert<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "data": {
        "totalConvertedAmount": 20,
        "totalCount": 2,
        "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
}
```

### 2.2.4 Get Free Game Players List

To get list of registered players of a free game.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGamePlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| free_game_id | Integer | Yes | Unique identity of free game |
| from_time | Long | Yes | Time range of free games' start times (Unix time stamp in milliseconds) |
| to_time | Long | Yes | |
| player_name | String | No | Unique identity of free game player<br>**Note: Fuzzy matching is implemented** |
| from_created_time | Long | No | Time range of when players are registered into free games (Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| free_game_name | String | No | Given name of free game |
| status | Integer | No | Status of free game wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/GetFreeGamePlayers?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:

```
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
egame2019&status=1
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of players in a free game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player’s free game wallet |
| playerId | Integer | Unique identity of player in free game |
| playerName | String | Unique identity of player in free game |
| walletType | String | Type of player wallet in free game:<br>G: Free Game<br>B: Bonus |
| balanceAmount | Decimal | Player balance of free game |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game. Refers to the total amount that need to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonusMaximumConversionAmount | Decimal | Maximum conversion amount for free game’s bonus |
| freeGameMaximumConversionAmount | Decimal | Maximum conversion amount for free game |
| freeGameCount | Integer | The player’s number of free games |
| status | Integer | Status of free game wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New |

| | | |
|---|---|---|
| | | 6: User Cancelled<br>8: Pending |
| expiredDate | Long | Expiry date of free game<br>(Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who created the free game |
| updatedBy | String | API or BackOffice user who updated the free game |
| updatedDate | Long | Date of when a player is registered or re-registered into the free game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Date of when a player is registered into the free game<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "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",
                "createdDate": 1546923615000,
                "updatedDate": 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",
                "createdDate": 1546918715000,
                "updatedDate": 1547028195000
            }
        ]
    },
    "error": null
}
```

### 2.2.5 Get List of Unregistered Players in Free Game

To get list of unregistered players in free game.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGamePendingPlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| free_game_id | Integer | Yes | Unique identity of free game |
| player_name | String | No | Unique identity of free game 's unregistered player<br>**Note: Fuzzy matching is implemented** |
| created_by | String | No | API or BackOffice user who registered the player |
| status | Integer | No | Status of wallet of unregistered player:<br>0: Inactive (Cancelled)<br>1: Pending<br>2: Transferred |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/GetFreeGamePendingPlayers?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_name=player1&free_game_id=12345&created_by=external&status=1
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of unregistered players in a free game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| pendingWalletId | Integer | Unique identity of free game's player's pending wallet |
| playerName | String | Unique identity of free game's un-registered player |
| status | Integer | Status of unregistered player's wallet:<br>0: Inactive (Cancelled)<br>1: Pending<br>2: Transferred |
| updatedDate | Long | Date of when a player is registered or re-registered into the free game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Date of when a player is registered into the free game<br>(Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who registered the player |
| updatedBy | String | API or BackOffice user who register or re-registered player |

Example:

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

### 2.2.6 Get All Player Details for Free Game

To get details of all players in a free game.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetFreeGameAllPlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| free_game_id | Integer | Yes | Unique identity of free game |
| from_time | Long | Yes | Time range of free games' start time (Unix time stamp in milliseconds) |
| to_time | Long | Yes | Time range of free games' end time (Unix time stamp in milliseconds) |
| player_name | String | No | Unique identity of free game's unregistered player<br>**Note: Fuzzy matching is implemented** |
| from_created_time | Long | No | Time range of free games' created time<br>(Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| free_game_name | String | No | Given name of free game |
| status | Integer | No | Status of free game wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed |

| | | | |
|---|---|---|---|
| | | | 5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/GetFreeGameAllPlayers?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
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
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of all players in a free game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player's free game wallet |
| playerId | Integer | Unique identity of free game's player |
| playerName | String | Unique identity of free game's player |
| walletType | String | Type of player wallet:<br>G: Free Game<br>B: Bonus |
| balanceAmount | Decimal | Player balance of unconverted free game, or converted amount of converted free game |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game.<br>Refers to the total amount that needs to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonusMinimumConversionAmount | Decimal | Minimum conversion amount for free game's bonus |
| bonusMaximumConversionAmount | Decimal | Maximum conversion amount for free game's bonus |
| freeGameMaximumConversionAmount | Decimal | Maximum conversion amount for free game |
| status | Integer | Status of free game wallet:<br>0: Inactive (Cancelled) |

| | | |
|---|---|---|
| | | 1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |
| expiredDate | Long | Expiry date of free game<br>(Unix time stamp in milliseconds) |
| updatedDate | Long | Date of when a player is registered or re-registered into the free game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Date of when a player is registered into the free game<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "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,
                "updatedDate": 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,
                "updatedDate": 1547028195000,
            }
        ]
    },
    "error": null
}
```

### 2.2.7 Get Single Player Details in Free Game

To get the details of a single player in a free game.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/GetPlayerFreeGames
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| player_name | String | Yes | Unique identity of free game's unregistered player |
| status | Integer Array | No | Status of free game wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:
```
https://api.pg-bo.me/external/FreeGame/v1/GetPlayerFreeGames?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&p
layer_name=player1&status=1&status=2
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of free games of a single player |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| freeGameId | Integer | Unique identity free game |
| freeGameName | String | Given name of free game |
| transactionId | String | Unique identity of transaction |
| gameIds | Integer Array | Unique identity of games |
| gameCount | Integer | Total number of given free games |
| totalGame | Integer | Total number of free games that will be given to a single player |
| balanceAmount | Decimal | Player balance of unconverted free game, or converted amount of converted free game |
| minimumConversionAmount | Decimal | Minimum conversion amount for free game |
| maximumConversionAmount | Decimal | Maximum conversion amount for free game |
| multiplier | Integer | Bet multiplier of free game |
| coinSize | Decimal | Coin size of free game |
| expiredDate | Long | Expiry date of free game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Created date of free game<br>(Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who created the free game |
| updatedBy | String | API or BackOffice user who updated the free game |
| isSupressDiscard | Boolean | Permission to allow players to cancel offer<br>True: Do not allow player to cancel offer<br>False: Allows player to cancel offer |
| status | Integer | Status of free game wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

| | | |
|---|---|---|
| conversionType | Char | Type of conversion after free game is completed:<br>C: Cash<br>B: Bonus |

Example:

```
{
    "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
}
```

### 2.2.8 Create Free Game

To create a new free game.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CreateFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | Yes | Currency of free game |
| free_game_name | String | Yes | Given name of free game |
| expired_date | long | Yes | Expire date of free game<br>(Unix time stamp in milliseconds) |
| bonus_minimum_conversion_amount | Decimal | No* | Minimum conversion amount for bonus<br>NULL: No limit for conversion amount |
| bonus_maximum_conversion_amount | Decimal | No* | Maximum conversion amount for bonus<br>NULL: No limit for conversion amount |
| minimum_conversion_amount | Decimal | No | Minimum conversion amount for free game<br>NULL: No limit for conversion amount |
| maximum_conversion_amount | Decimal | No | Maximum conversion amount for free game<br>NULL: No limit for conversion amount |
| bonus_type | Integer | No* | Bonus wagering condition<br>0: Bonus<br>1: Cash<br>2: Bonus & Cash |
| bonus_ratio_amount** | Decimal | No* | Wagering requirement for bonus game. |

| | | | |
|---|---|---|---|
| | | | Refers to the amount that needs to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonus_ratio** | Decimal | No* | Wagering requirement for bonus game. Refers to the multiple of the turnover in free game that needs to be completed after the amount won in free game has been transferred to the bonus wallet<br>**Turnover = bonus ratio x converted bonus amount** |
| conversion_type | Char | Yes | Type of conversion after free game is completed:<br>C: Cash<br>B: Bonus |
| multiplier | Integer | Yes | Bet multiplier of free game |
| coin_size | Decimal | Yes | Coin size of free game |
| game_count | Integer | Yes | Total number of free games that will be given to each player |
| game_ids | Integer Array | Yes | Unique identity of games<br>**(Note: Only games with same base bet are accepted)** |
| is_event | Boolean | Yes | Type of free game:<br>True: Flexible<br>False: Standard |
| player_name | String | No | Unique identity of player. System will create free game and assign to this player directly<br>**Note: Free game type will be changed to flexible mode if player_name is provided** |
| transaction_id | String | Yes | Unique identity of transaction.<br>**Note: System will return a corresponding free game id for a duplicated transaction** |
| non_discardable | Boolean | No | To allow player to discard the offer:<br>True: Not Allow player to discard the offer<br>False: Allow player to discard the offer |

*Only applicable for bonus conversion type
**Turnover condition of *bonus_ratio* will be overridden if *bonus_ratio amount* is provided

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/CreateFreeGame?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&free_game_name=F
reeGame2019&expired_date=1745289243000&status=1&bonus_maximum_conversion_amount=10
&maximum_conversion_amount=1&bonus_type=1&bonus_ratio_amount=2&conversion_type=B&m
ultiplier=5&coin_size=20&game_count=20&game_ids=1&game_ids=2&is_event=true&player_
name=player1&transaction_id=12345&non_discardable=False
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| freeGameId | Integer | Unique identity of free game |

Example:

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

### 2.2.9 Create Free Game by Bet Amount

To create a new free game using the desired bet amount.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CreateFreeGameByBetAmount
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | Yes | Currency of free game |
| free_game_name | String | Yes | Given name of free game |
| expired_date | long | Yes | Expiry date of free game<br>(Unix time stamp in milliseconds) |
| bonus_minimum_conversion_amount | Decimal | No* | Minimum conversion amount for bonus<br>NULL: No limit for conversion amount |
| bonus_maximum_conversion_amount | Decimal | No* | Maximum conversion amount for bonus<br>NULL: No limit for conversion amount |
| minimum_conversion_amount | Decimal | No | Minimum conversion amount for free game<br>NULL: No limit for conversion amount |
| maximum_conversion_amount | Decimal | No | Maximum conversion amount for free game<br>NULL: No limit for conversion amount |
| bonus_type | Integer | No* | Bonus wagering condition:<br>0: Bonus<br>1: Cash<br>2: Bonus & Cash |
| bonus_ratio_amount** | Decimal | No* | Wagering requirement for bonus game. |

| | | | |
|---|---|---|---|
| | | | Refers to the amount that needs to be completed after the amount won in free game has been transferred to the bonus wallet |
| bonus_ratio** | Decimal | No* | Wagering requirement for bonus game. Refers to the multiple of the turnover in free game that needs to be completed after the amount won in free game has been transferred to the bonus wallet<br>**Turnover = bonus ratio x converted bonus amount** |
| conversion_type | Char | Yes | Type of conversion for after free game is completed:<br>C: Cash<br>B: Bonus |
| bet_amount | Decimal | Yes | Desired bet amount for each game.<br>**Note: System will round down to closest multiplier and coin size automatically** |
| game_count | Integer | Yes | Total number of free games that will be given to each player |
| game_ids | Integer Array | Yes | Unique identity of games<br>**(Note: Only games with same base bet are accepted)** |
| is_event | Boolean | Yes | Type of free game:<br>True: Flexible<br>False: Standard |
| player_name | String | No | Unique identity of player. System will create free game and assign to this player directly<br>**Note: Free game type will be changed to flexible mode if player_name is provided** |
| transaction_id | String | Yes | Unique identity of transaction.<br>**Note: System will return a corresponding free game ID for a duplicate transaction** |
| non_discardable | Boolean | No | To allow player to discard the offer:<br>True: Not Allow player to discard the offer<br>False: Allow player to discard the offer |

*Only applicable for bonus conversion type
**Turnover condition of *bonus_ratio* will be overridden if *bonus_ratioamount* is provided

Example:

URL:

```
https://api.pg-
bo.me/external/FreeGame/v1/CreateFreeGameByBetAmount?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&free_game_name=F
reeGame2019&expired_date=1745289243000&status=1&bonus_maximum_conversion_amount=10
&maximum_conversion_amount=1&bonus_type=1&bonus_ratio_amount=2&conversion_type=B&b
et_amount=0.6&game_count=20&game_ids=1&game_ids=2&is_event=true&player_name=player
1&transaction_id=12345&non_discardable=False
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| freeGameId | Integer | Unique identity of free game |

Example:

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

## 2.2.10 Transfer Player to Standard Free Game

To add players into a standard free game.

### Notes

- We suggest do not exceed 200 players per request.

### Request

API URL : {PgSoftAPIDomain}/FreeGame/v1/TransferInFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| free_game_id | Integer | Yes | Unique identity of free game |
| player_names | String Array | Yes | Unique identity of players |
| transfer_reference | String | Yes | Unique identity of transfer |

Example:

```
URL:
https://api.pg-bo.me/external/FreeGame/v1/TransferInFreeGame?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311

Body:
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&free_game_id=12&player_names=
player1&player_names=player2&transfer_reference=freegame_20181128
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | String | Unique identity of transaction |
| operatorToken | String | Unique identity of operator |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| status | Integer | Status of transfer request:<br>0: Failed<br>1: Successful<br>2: Pending (For non-existent player, free game will be assigned to the player automatically after player is created.) |

Example:

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

## 2.2.11 Transfer Player to Flexible Free Game

To add players into a flexible free game. Operator can override the free game setting for a specific player.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/TransferInFlexibleFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_free_games | JSON String Array | Yes | The free game details that have been overridden for a single player |
| free_game_id | Integer | Yes | Unique identity of free game |
| transfer_reference | String | Yes | Unique identity of transfer |
| allow_multiple | Boolean | Yes | Allows player to re-register<br>**True**: No. of times a player can re-register in a free game<br>**False**: Player is only allowed to register once in a free game. |

player_free_games JSON String:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| player_name | String | Yes | Unique identity of operator |
| free_game_count | Integer | Yes | Free game count that can be overridden for player<br>**Value range: Value ≥ 1** |

| | | | |
|---|---|---|---|
| is_unlimited_bonus_maximum_conversion_amount | Bool | Yes | Sets bonus maximum conversion amount to unlimited |
| is_unlimited_free_game_maximum_conversion_amount | Bool | Yes | Sets free game maximum conversion amount to unlimited |
| bonus_minimum_conversion_amount | Decimal | No* | Bonus minimum conversion amount that can be overridden for a player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amouvalue = 0** |
| bonus_maximum_conversion_amount | Decimal | No* | Bonus maximum conversion amount that can be overridden for a player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amount if value = 0** |
| free_game_minimum_conversion_amount | Decimal | No** | Free game minimum conversion amount that can be overridden for a player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amount if value = 0** |
| free_game_maximum_conversion_amount | Decimal | No** | Free game maximum conversion amount that can be overridden for a player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amount if value = 0** |
| description | String | No | Reference note for transfer |

*Only applicable if *is_unlimited_bonus_maximum_conversion_amount* parameter value is false
**Only applicable if *is_unlimited_free_game_maximum_conversion_amount* parameter value is false

Example:

URL:

```
https://api.pg-
bo.me/external/FreeGame/v1/TransferInFlexibleFreeGame?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:

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

**Response**

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | String | Unique identity of transaction |
| operatorToken | String | Unique identity of operator |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| freeGameCount | Integer | Number of player free games |
| status | Integer | Status of transfer request:<br>0: Failed<br>1: Successful<br>2: Pending (For non-existent player, free game will be assigned to the player automatically after player is created.) |

Example:

```
{
    "data": [
        {
            "transactionId": 10,
            "operatorToken": "abc123",
            "playerName": "player1",
            "currencyCode": "CNY",
            "freeGameCount": 10,
            "status": 1
        } ,
        {
            "transactionId": 1,
            "operatorToken": "1234",
            "playerName": "player2",
            "currencyCode": "CNY",9
            "freeGameCount": 10,
            "status": 1
        }
    ],
    "error": null
}
```

## 2.2.12 Cancel Free Game

To cancel a created free game.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| free_game_id | String | Yes | Unique identity of free game |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/CancelFreeGame?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&free_game_id=10919
```

**Response**

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

### 2.2.13 Cancel Free Game of a Player

To cancel a free game of a specific player.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPlayerFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of player |
| balance_id | Integer | Yes | Unique identity of player's free game wallet |

Example:

URL:
```
https://api.pg-bo.me/external/FreeGame/v1/CancelPlayerFreeGame?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&balance_i
d=10919
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

### 2.2.14 Cancel Free Game of an Unregistered Player

To cancel a free game for a specific unregistered player.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPendingPlayerFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| pending_wallet_id | Integer | Yes | Unique identity of player's free game wallet |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/CancelPendingPlayerFreeGame?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&pending_wallet_id=10919
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

### 2.2.15 Cancel All Free Games of a Player

To cancel all free games of a player.

**Request**

API URL : {PgSoftAPIDomain}/FreeGame/v1/CancelPlayerAllFreeGame
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of player |
| free_game_id | Integer | Yes | Unique identity of free game |

Example:

URL:

```
https://api.pg-bo.me/external/FreeGame/v1/CancelPlayerAllFreeGame?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&free_game_id=10919
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>True: Successful |

Example:

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

## 2.2.16 Get Free Game Player Summary

To get free game player summary.

**Request**

API URL : {DataGrabAPIDomain}/FreeGame/v1/GetFreeGamePlayerSummary
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| row_version | Long | No | Updated time of data<br>(Unix time stamp in milliseconds) |
| player_name | String | No | Unique identity of player |
| row_count | Integer | No | Number of records for each batch<br>**Note:**<br>• **Default value: 1**<br>• **Value range: 1-5000** |
| currency | String | No | Currency of player |
| free_game_id | Integer | No | Unique identity of free game |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/
FreeGame/v1/GetFreeGamePlayerSummary?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&row_version=1346592723000&pla
yer_name=testplayer1&row_count=10&currency=CNY&free_game_id=12345
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| lastRowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| result | JSON Array | List of the result |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| freeGameId | Ulong | Unique identity of free game Id |
| freeGameParentId | Ulong | Unique identity of free game parent Id |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| totalFreeGame | Integer | Total number of given free games |
| coinSize | Decimal | Bet size of game |
| multiplier | Integer | Bet multiplier of free game |
| payout | Decimal | Convert to cash amount of free game |
| remainCount | Integer | Total number of remain free games |
| conversionType | Char | Type of conversion after free game is completed:<br>C: Cash<br>B: Bonus |
| status | Integer | Status of free game:<br>0: Inactive<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |
| createdTime | Long | Free game created time |
| updatedTime | Long | Free game updated time |

Example:

```
{
    "data": {
        "lastRowVersion": 1526371744000,
        "result": [
            {
                "freeGameId": 2,
                "freeGameParentId": 2,
                "playerName": "qa1",
                "currencyCode": "CNY",
                "totalFreeGame": 1,
                "coinSize": 0.01,
                "multiplier": 10,
                "payout": 0.0,
                "remainCount": 1,
                "conversionType": "Bonus",
                "status": 0,
                "createdTime": 1526371189000,
                "updatedTime": 1526371744000
            }
        ]
    },
    "error": null
}
```

## 2.3 Bonus [Latest version: v1]

### 2.3.1 Get Bonus Game List

To get the created bonus games list.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| from_time | Long | Yes | Time range of bonus games’ start times<br>(Unix time stamp in milliseconds) |
| to_time | Long | Yes | |
| from_created_time | Long | No | Time range of bonus games’ created times<br>(Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| bonus_id | Integer | No | Unique identity of bonus game |
| currency | String | No | Currency of bonus game |
| bonus_name | String | No | Given name of bonus game |
| bonus_transfer_type | Integer | No | Transfer type for bonus game:<br>0: Standard<br>3: Flexible |
| status | Integer | No | Status of bonus game:<br>0: Inactive<br>1: Active<br>2: Expired |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/GetBonus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
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
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of details of multiple bonus games |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorToken | String | Unique identity of operator |
| bonusId | Integer | Unique identity of bonus game |
| bonusName | String | Given name of bonus game |
| isEvent | Boolean | Type of bonus game:<br>True: Flexible<br>False: Standard |
| gameIds | Integer Array | Unique identity of games |
| currencyCode | String | Currency of bonus game |
| bonusRatio | Decimal | Wagering requirement for bonus game.<br>Refers to the multiple of the turnover that needs to be completed |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game.<br>Refers to the total amount that needs to be completed |
| bonusMinimumConversionAmount | Decimal | The minimum cash conversion amount of free game |
| bonusMaximumConversionAmount | Decimal | The maximum cash conversion amount of free game |
| bonusType | Integer | Bonus game wagering condition:<br>0: Bonus<br>1: Cash<br>2: Bonus & Cash |
| status | Integer | Status of bonus game: |

| | | |
|---|---|---|
| | | 0: Inactive (Cancelled)<br>1: Active<br>2: Expired |
| bonusAmountGiven | Decimal | Total amount of bonus given |
| numberOfPlayers | Integer | Total number of players in bonus game |
| expiredDate | Long | Expiry date of bonus game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | Created date of bonus game<br>(Unix time stamp in milliseconds) |
| isSuppressDiscard | Boolean | Permission to allow players to cancel offer<br>True: Do not allow player to cancel offer<br>False: Allows player to cancel offer |

Example:

```
{
    "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
}
```

## 2.3.2 Get List of Bonus Games Converted to Cash

To get list of bonus games that has been converted to cash.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusConvertedWallet
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| bonus_id | Integer | Yes | Unique identity of bonus |
| from_transaction_time | Long | Yes | Time range of bonus player's transactions<br>(Unix time stamp in milliseconds) |
| to_transaction_time | Long | Yes | |
| player_name | String | No | Unique identity of player<br>**Note: Fuzzy matching is implemented** |

Example:

URL:
```
https://api.pg-bo.me/external/Bonus/v1/GetBonusConvertedWallet?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&b
onus_id=12345&player_name=player1&from_transaction_time=1346592723000&to_transacti
on_time=1346592723000
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalConvertedAmount | Decimal | Total amount that has been converted to cash through the bonus game |
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of players details whose bonus game has been converted to cash |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player’s bonus game wallet |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| gameId | Integer | Unique identity of game |
| transactionAmount | Decimal | Converted amount of player |
| transactionDateTime | Long | Time of conversion<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "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
}
```

### 2.3.3 Get Bonus Game Players List

To get the list of registered players of a bonus game.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusPlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| bonus_id | Integer | Yes | Unique identity of bonus |
| from_time | Long | Yes | Time range of bonus games’ start times (Unix time stamp in milliseconds) |
| to_time | Long | Yes | |
| player_name | String | No | Unique identity of bonus player<br>**Note: Fuzzy matching is implemented** |
| from_created_time | Long | No | Time range of when players are registered into bonus games (Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| status | Integer | No | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/GetBonusPlayers?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
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
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of players in bonus game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player’s bonus wallet |
| playerId | Integer | Unique identity of bonus game’s player |
| playerName | String | Unique identity of bonus game’s player |
| walletType | String | Type of player wallet in bonus game:<br>G: Free Game<br>B: Bonus |
| balanceAmount | Decimal | Player’s balance in bonus game |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game.<br>Refers to the total amount that needs to be completed |
| MinimumConversionAmount | Decimal | Minimum conversion amount for bonus |
| MaximumConversionAmount | Decimal | Maximum conversion amount for bonus |
| status | Integer | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |
| expiredDate | Long | Expiry date of bonus game<br>(Unix time stamp in milliseconds) |

| | | |
|---|---|---|
| updatedDate | Long | Date of when a player is registered or re-registered into the bonus game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | |

Example:

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

### 2.3.4 Get List of Unregistered Players of Bonus Game

To get list of unregistered players of a bonus game.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusPendingPlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| bonus_id | Integer | Yes | Unique identity of bonus game |
| player_name | String | No | Unique identity of bonus game's unregistered player<br>**Note: Fuzzy matching is implemented** |
| created_by | String | No | API or BackOffice user who registered the player |
| status | Integer | No | Status of unregistered player's wallet:<br>0: Inactive (Cancelled)<br>1: Pending<br>2: Transferred |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/GetBonusPendingPlayers?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&p
layer_name=player1&bonus_id=12345&created_by=external&status=1
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of unregistered players in a bonus game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| pendingWalletId | Integer | Unique identity of unregistered player’s bonus wallet |
| playerName | String | Unique identity of unregistered player in bonus game |
| status | Integer | Status of unregistered player’s wallet:<br>0: Inactive (Cancelled)<br>1: Pending<br>2: Transferred |
| updatedDate | Long | Date of when a player is registered or re-registered into the bonus game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | |
| createdBy | String | API or BackOffice user who registered the player |
| updatedBy | String | |

Example:

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

### 2.3.5 Get All Player Details in Bonus Game

To get the details of all players in a bonus game.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetBonusAllPlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| bonus_id | Integer | Yes | Unique identity of bonus game |
| from_time | Long | Yes | Time range of bonus games' start times<br>(Unix time stamp in milliseconds) |
| to_time | Long | Yes | |
| player_name | String | No | Unique identity of bonus game's player<br>**Note: Fuzzy matching is implemented** |
| from_created_time | Long | No | Time range of bonus games' created times<br>(Unix time stamp in milliseconds) |
| to_created_time | Long | No | |
| status | Integer | No | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/GetBonusAllPlayers?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
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&bonus_id=12345&status=1
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of all players in a bonus game |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| balanceId | Integer | Unique identity of player's bonus wallet |
| playerId | Integer | Unique identity of bonus game's player |
| playerName | String | Unique identity of bonus game's player |
| walletType | String | Type of player wallet:<br>G: Free Game<br>B: Bonus |
| balanceAmount | Decimal | Player balance of unconverted bonus game, or converted amount of converted bonus game |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game. Refers to the total amount that needs to be completed |
| MinimumConversionAmount | Decimal | Minimum conversion amount for bonus game |
| MaximumConversionAmount | Decimal | Maximum conversion amount for bonus game |
| status | Integer | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |
| expiredDate | Long | Expiry date of bonus game |

| | | |
|---|---|---|
| | | (Unix time stamp in milliseconds) |
| updatedDate | Long | Date of when a player is registered or re-registered into the bonus game<br>(Unix time stamp in milliseconds) |
| createdDate | Long | |

Example:

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

### 2.3.6 Get Single Player Details in Bonus Game

To get the details of a single player in a bonus game.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/GetPlayerBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| player_name | String | Yes | Unique identity of bonus player |
| status | Integer Array | No | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/GetPlayerBonus?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&p
layer_name=player1&status=1&status=1&status=2
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of bonus games of a single player |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| bonusId | Ulong | Unique identity of bonus game<br>**Note: Free game id will be returned for bonus that have been converted from free game** |
| bonusName | String | Given name of bonus game |
| gameIds | Integer Array | Unique identity of games |
| balanceAmount | Decimal | Player balance of unconverted bonus game, or converted amount of converted bonus game |
| maximumConversionAmount | Decimal | Maximum conversion amount for bonus |
| bonusRatioAmount | Decimal | Wagering requirement for bonus game.<br>Refers to the total amount that needs to be completed |
| bonusParentType | Integer | Type of bonus parent:<br>B: Bonus game that was created normally<br>G: Bonus game that was converted from free game |
| status | Integer | Status of bonus wallet:<br>0: Inactive (Cancelled)<br>1: Active<br>2: Expired<br>3: Converted<br>4: Completed<br>5: New<br>6: User Cancelled<br>8: Pending |
| createdDate | Long | Created date of bonus game<br>(Unix time stamp in milliseconds) |
| expiredDate | Long | Expiry date of bonus game<br>(Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who created or updated the bonus game |
| updatedBy | String | |

Example:

```
{
    "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
}
```

## 2.3.7 Create Bonus Game

To create new bonus game.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/CreateBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | Yes | Currency of bonus game |
| bonus_name | String | Yes | Given name of bonus game |
| bonus_amount | decimal | Yes | Bonus amount to be given for each player |
| expired_date | long | Yes | Expiry date of bonus game<br>(Unix time stamp in milliseconds) |
| minimum_conversion_amount | Decimal | No | Minimum conversion amount for bonus game<br>NULL: No limit for conversion amount |
| maximum_conversion_amount | Decimal | No | Maximum conversion amount for bonus game<br>NULL: No limit for conversion amount |
| bonus_type | Integer | Yes | Bonus game wagering condition:<br>0: Bonus<br>1: Cash<br>2: Bonus & Cash |
| bonus_ratio_amount** | Decimal | No | Wagering requirement for bonus game. Refers to the total amount that needs to be completed |

| | | | |
|---|---|---|---|
| bonus_ratio** | Decimal | Yes | Wagering requirement for bonus game. Refers to the multiple of the turnover that needs to be completed<br>**Turnover = bonus ratio x bonus amount** |
| game_ids | Integer Array | Yes | Unique identity of games<br>**Note: Only games with same base bet are accepted** |
| is_event | Boolean | Yes | Type of bonus game:<br>True: Flexible<br>False: Standard |
| player_name | String | No | Unique identity of player. System will create bonus game and assign to this player directly<br>**Note: Free game type will be changed to flexible mode if player_name is provided** |
| transaction_id | String | Yes | Unique identity of transaction.<br>**Note: System will return a corresponding bonus ID for a duplicated transaction** |
| non_discardable | Boolean | No | To allow player to discard the offer:<br>True: Not Allow player to discard the offer<br>False: Allow player to discard the offer |

**Turnover condition of *bonus_ratio* will be overridden if *bonus_ratio_amount* is provided

Example:
`URL:`
`https://api.pg-bo.me/external/Bonus/v1/CreateBonus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311`

`Body:`
```
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&player_name=player1&transaction_id=12345&non_discardable=False
```

## Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| **bonusId** | Integer | Unique identity of bonus game |

Example:

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

### 2.3.8 Transfer Player to Standard Bonus Game

To add players into a created standard bonus game. Operator can only transfer players into a pre-set bonus game.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/TransferInBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| bonus_id | Integer | Yes | Unique identity of bonus game |
| player_names | String Array | Yes | Unique identity of players |
| transfer_reference | String | Yes | Unique identity of transfer |

Example:

URL:
```
https://api.pg-bo.me/external/Bonus/v1/TransferInBonus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&bonus_id=12&player_names=player1&player_names=player2&transfer_reference=bonusgame_20181128
```

**Response**

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | String | Unique identity of transaction |
| operatorToken | String | Unique identity of operator |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| status | Integer | Status of transfer request:<br>0: Failed<br>1: Successful<br>2: Pending (For non-existent player, bonus game will be assigned to the player automatically after player is created.) |

Example:

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

### 2.3.9 Transfer Player to Flexible Bonus Game

To add players into a created flexible bonus game. Operator can override bonus game setting for a specific player.

**Notes**

- We suggest do not exceed 200 players per request.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/TransferInFlexibleBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_bonuses | JSON String Array | Yes | Overridden default bonus game's information for players |
| bonus _id | Integer | Yes | Unique identity of bonus game |
| transfer_reference | String | Yes | Unique identity of transfer |
| allow_multiple | Boolean | Yes | **True**: No. of times a player can re-register in a bonus game<br>**False**: Player is only allowed to register once in a bonus game. |

player_bonuses JSON String:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| player_name | String | Yes | Unique identity of operator |
| bonus_amount | Integer | Yes | Custom bonus amount for player<br>**Value range: Value ≥ 1** |
| is_unlimited_bonus_maximum_conversion_amount | Boolean | Yes | Sets bonus game's maximum conversion amount to unlimited |

| | | | |
|---|---|---|---|
| bonus_minimum_conversion_amount | Decimal | No* | Custom bonus game’s minimum conversion amount for player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amount if value = 0** |
| bonus_maximum_conversion_amount | Decimal | No* | Custom bonus game’s maximum conversion amount for player<br>**Value range: Value ≥ 1**<br>**Default value: 0**<br>**Note: System will use default conversion amount if value = 0** |
| description | String | No | Reference note for transfer |

*Only applicable if *is_unlimited_bonus_maximum_conversion_amount* parameter value is false

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/TransferInFlexibleBonus?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:

```
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_
refer
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | String | Unique identity of transaction |
| operatorToken | String | Unique identity of operator |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| status | Integer | Status of transfer request:<br>0: Failed<br>1: Successful<br>2: Pending (For non-existent player, bonus game will be assigned to the player automatically after player is created) |

Example:

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

## 2.3.10 Cancel Bonus Game

To cancel a created bonus game.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| bonus _id | String | Yes | Unique identity of bonus game |

Example:

URL:
```
https://api.pg-bo.me/external/Bonus/v1/CancelBonus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&bonus_id=10919
```

**Response**

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

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

## 2.3.11 Cancel Bonus Games of Registered Player

To cancel a bonus game of a specific registered player.

**Request**

API URL : {PgSoftAPIDomain}/Bonus/v1/CancelPlayerBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of player |
| balance_id | Integer | Yes | Unique identity of player's bonus wallet |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/CancelPlayerBonus?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&balance_i
d=10919
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

## 2.3.12 Cancel Bonus Game of Unregistered Player

To cancel a bonus game of a specific unregistered player.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| pending_wallet_id | Integer | Yes | Unique identity of player's bonus wallet |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/CancelPendingPlayerBonus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&pending_wallet_id=10919
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

## 2.3.13 Cancel All Bonus Games of a Player

To cancel all bonus games of a player.

### Request

API URL : {PgSoftAPIDomain}/Bonus/v1/CancelPlayerAllBonus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of player |
| bonus_id | Integer | Yes | Unique identity of bonus game |

Example:

URL:

```
https://api.pg-bo.me/external/Bonus/v1/CancelPlayerAllBonus?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&bonus_id=
10919
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Request results:<br>False: Failed<br>True: Successful |

Example:

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

## 2.4 Tournament [Latest version: v2]

### 2.4.1 Get Tournaments List

To get list of tournaments.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournaments
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| from_created_time | Long | Yes | Time range of tournament's start time and end time<br>(Unix time stamp in milliseconds) |
| to_created_time | Long | Yes | |
| status | Integer Array | No | Status of tournament:<br>0: Cancel<br>1: InProgress<br>2: New<br>3: End |
| game_ids | Integer Array | No | Unique identity of games |
| language | String | No | Response language for tournament information:<br>en-US: English (Default)<br>zh-CN: Chinese |
| start_time | Long | No | Time range of tournaments' start times and end times<br>(Unix time stamp in milliseconds) |
| end_time | Long | No | |
| currency | string | No | Currency used in the tournament |

Example:

URL:

```
https://api.pg-bo.me/external/Tournament/v2/GetTournaments?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&status=1&status=2&game_ids=1&
game_ids=2&language=zh-
cn&start_time=1559145500000&end_time=1559145600000&currency=CNY&from_created_time=
1559145500000&to_created_time=1559145600000
```

### Response

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| tournamentKey | String | Unique key of tournament |
| tournamentId | Long | Unique identity of tournament |
| subTournamentId | Long | Unique identity of Sub-Tournament |
| name | String | Name of tournament |
| description | String | Description of tournament |
| registrationFees | String | Registration fee of tournament |
| prize_type | Integer | Type of Tournament prize<br>0: Item prize<br>1: Cash prize |
| prize | String Array | Prizes of tournament |
| registrationUrl | String | Registration URL of tournament |
| hasSubTournaments | Boolean | To indicate if Sub-Tournament exists |
| subTournamentStatus | Decimal | Status of Sub-Tournament |
| subTournamentStartTime | Long | Start time of Sub-Tournament<br>(Unix time stamp in milliseconds) |
| subTournamentEndTime | Long | End time of Sub-Tournament<br>(Unix time stamp in milliseconds) |
| numberOfPlayers | Integer | Number of participants in tournament |
| initialPoint | Decimal | Initial balance of tournament |
| qualifiedPoint | Decimal | Points needed for tournament player to enter the leaderboard |
| gameMode | Integer | Tournament mode:<br>1: Accumulation<br>2: Depletion<br>3: Hybrid (Accumulation + Depletion) |
| rankStatus | Integer | Tournament leaderboard status:<br>0: Disabled<br>1: Enabled<br>2: Cut off (Tournament rank had been finalized) |

| | | |
|---|---|---|
| status | Integer | Status of Tournament:<br>0: Cancel<br>1: InProgress<br>2: New<br>3: End |
| startTime | Long | Start time of tournament<br>(Unix time stamp in milliseconds) |
| endTime | Long | End time of tournament<br>(Unix time stamp in milliseconds) |
| createdTime | Long | Created time or updated time of tournament<br>(Unix time stamp in milliseconds) |
| updatedTime | Long | |
| isCashMode | Boolean | Tournament Mode:<br>Cash Tournament<br>Point Tournament |
| isRequiredRegister | Boolean | To allow players to join the tournament manually or automatically |
| group | String Array | List of Currency |
| gameIds | Uint Array | List of game IDs |
| turnOverLimit | Decimal | Points required for players to join the tournament (total amount the player has bet or spent) |
| createdBy | String | API or BackOffice user who created tournament |
| operatorInfos | JSON Array | Operator details |

operatorInfos JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorName | String | Name of operator |
| operatorToken | String | Unique identity of operator |
| group | String Array | Currency list |
| createdTime | Long | Created time of tournament |
| updatedTime | Long | Last updated time of tournament |
| createdBy | String | Creator of tournament |
| updatedBy | String | Last update source of tournament |

Example:

```
{
    "data": [{
                "tournamentKey": "1857-2270-0_P",
                "tournamentId": 1857,
                "name": "Point Testing",
                "operatorToken": "a**d",
                "description": "",
                "gameIds": [
                    94
                ],
                "registrationFees": "",
                "prizeType": 0,
                "prize": {
                    "winnerPrizeDetails": [
                        {
                            "positionTo": 1,
                            "item": "3"
                        },
                        {
                            "positionTo": 2,
                            "item": "2"
                        },
                        {
                            "positionTo": 3,
                            "item": "1"
                        }
                    ]
                },
                "registrationUrl": "",
                "hasSubTournaments": false,
                "numberOfPlayers": 1,
                "initialPoint": 1.00000,
                "qualifiedPoint": 2.00000,
                "turnoverLimit": 1.0,
                "gameMode": 1,
                "rankStatus": 2,
                "isCashMode": false,
                "status": 3,
                "startTime": 1607558400000,
                "endTime": 1607644799000,
                "subTournamentId": 2270,
                "subTournamentStatus": 3,
                "subTournamentStartTime": 1607558400000,
                "subTournamentEndTime": 1607644799000,
                "createdTime": 1607569038000,
                "updatedTime": 1607569038000,
                "createdBy": "ExternalAPI",
                "isRequiredRegister": false
        },
       {
            "tournamentKey": "1859-2272-0_P",
            "tournamentId": 1859,
            "name": "TournamentName",
            "operatorToken": "a**d",
            "description": "TournamentDesc",
            "group": [
                "JPY"
            ],
            "gameIds": [
                1,
                2
            ],
            "registrationFees": "RegistrationFees",
            "prizeType": 0,
            "prize": {
                "winnerPrizeDetails": [
                    {
                        "positionTo": 1,
                        "item": "1st prize"
                    },
                    {
                        "positionTo": 3,
                        "item": "2rd prize"
                    },
                    {
                        "positionTo": 10,
                        "item": "3rd prize"
                    }
                ]
            },
            "registrationUrl": "RegistrationUrl",
            "hasSubTournaments": false,
            "numberOfPlayers": 0,
            "initialPoint": 200.00000,
            "qualifiedPoint": 200.00000,
            "turnoverLimit": 0.0,
            "gameMode": 1,
            "rankStatus": 2,
            "isCashMode": false,
            "status": 3,
            "startTime": 1607644800000,
            "endTime": 1607731200000,
            "subTournamentId": 2272,
            "subTournamentStatus": 3,
            "subTournamentStartTime": 1607644800000,
            "subTournamentEndTime": 1607731200000,
            "createdTime": 1607680643000,
            "updatedTime": 1607680643000,
            "createdBy": "External API",
            "isRequiredRegister": false
        }],
    "error": null
}
```

## 2.4.2 Get Tournament Players List

To get list of tournament players.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentPlayersAsync
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| sub_tournament_id | Integer | Yes | Unique identity of Sub-Tournament |
| page_number | Integer | Yes | Batch number of data return |
| row_count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1-5000** |
| player_names | String | No | Unique identity of tournament players |
| from_created_time | Long | No | Time range of when players were added into a tournament.<br>(Unix time stamp in milliseconds) |
| to_created_time | Long | No | |

Example:

URL:

```
https://api.pg-
bo.me/external/Tournament/v2/GetTournamentPlayersAsync?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
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
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPage | Integer | Total number of available batches |
| result | JSON Array | List of players in tournament |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorToken | String | Unique identity of operator |
| tournamentPlayerId | String | Unique identity of tournament player |
| playerName | String | Unique identity of player |
| balanceAmount | Decimal | Remaining balance of a player in tournament |
| point | Decimal | Total points won by player in tournament |
| remainingRevive | Integer | Remaining times to revive in tournament |
| remainingRegister | Integer | Remaining times to re-register in tournament |
| createdTime | Long | Time when player was added into tournament (Unix time stamp in milliseconds) |
| updatedTime | Long | Time of when tournament player's information was updated (Unix time stamp in milliseconds) |
| createdBy | String | API or BackOffice user who registered the player into the tournament |

Example:

```
{
    "data": {
        "totalCount": 1,
        "result": [
            {
                "tournamentPlayerId": 0-456,
                "operatorToken": "1234",
                "playerName": "qafmcny18",
                "balanceAmount": 96,
                "point": 544.05,
                "remainingRevive": 0,
                "remainingRegister": 9,
                "createdTime": 1561640502000,
                "updatedTime": 1561640502000,
                "createdBy": "mei"
            },
             {
                "tournamentPlayerId": 0-456,
                "operatorToken": "1234",
                "playerName": "qafmcny19",
                "balanceAmount": 96,
                "point": 544.05,
                "remainingRevive": 0,
                "remainingRegister": 9,
                "createdTime": 1561640502000,
                "updatedTime": 1561640502000,
                "createdBy": "mei"
            }
        ],
        "totalPage": 1
    },
    "error": null
}
```

### 2.4.3 Get Tournament Top Rankings

To get tournament top rankings.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentTopRankingsAsync
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| tournament_id | Long | Yes | Unique identity of tournament |
| count | Integer | Yes | Records count of rankings<br>**Note: Value range 1-5000** |
| player_name | String | No | Unique identity of player |

Example:

```
URL:
https://api.pg-
bo.me/external/Tournament/v2/GetTournamentTopRankingsAsync?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311

Body:
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&tournament_id=41&count=10&pla
yer_name=player1
```

**Response**

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| top_ranking | JSON Array | List of tournament's top rankings |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorToken | Integer | Unique identity of operator |
| tournamentPlayerId | String | Unique identity of tournament player |
| playerName | String | Unique identity of player |
| point | Decimal | Total points won by player in tournament |
| totalSpendPoint | Decimal | Total points spent by player |
| previousRank | Integer | Previous ranking of player |
| rank | Integer | Current ranking of player |
| lastUpdatedOn | Long | Last updated time of leader board (Unix time stamp in milliseconds) |
| IsRankingQualifiedPlayer | Boolean | To identify if a player is qualified to join the tournament |

Example:

```
{
    "data":{
        "top_ranking": [
            {
                "tournamentPlayerId": 0-456,
                "operatorToken": "1234",
                "playerName": "player1",
                "point": 30360.98,
                "totalSpendPoint": 69376.8,
                "previousRank": 1,
                "rank": 1,
                "lastUpdatedOn": 1540380199000,
                "IsRankingQualifiedPlayer": true
            },
            {
                "tournamentPlayerId": 0-457,
                "operatorToken": "1234",
                "playerName": "player2",
                "point": 10000.74,
                "totalSpendPoint": 33,
                "previousRank": 2,
                "rank": 2,
                "lastUpdatedOn": 1540380199000
               "IsRankingQualifiedPlayer": true
            }
        ]
    },
    "error": null
}
```

### 2.4.4 Register Tournament Players

To add multiple players into a tournament.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/CreateTournamentPlayersAsync
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| tournament_id | Long | Yes | Unique identity of tournament<br>Note: Please create tournament through PG BackOffice |
| player_names | String Array | Yes | Unique identity of player |
| allow_re-register | Boolean | Yes | To allow player to re-register to the same tournament:<br>True: Allow<br>False: Disallow |

Example:

<u>URL:</u>

```
https://api.pg-bo.me/external/Tournament/v2/CreateTournamentPlayersAsync?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

<u>Body:</u>

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&tournament_id=41&player_names=player1 &player_names=player2&allow_re-register=true
```

**Response**

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| operatorToken | Integer | Unique identity of operator |
| tournamentPlayerId | String | Unique identity of tournament player |
| playerName | String | Unique identity of player |
| balanceAmount | Integer | Initial points in player balance |
| registerAllowed | Integer | Number of times allowed to re-register in tournament |
| reviveAllowed | Integer | Number of times allowed to revive in tournament |
| status | Integer | Status of Tournament Player:<br>0: Duplicated<br>1: Successful<br>3: Registration limit exceeded<br>4: Invalid Player |
| tournamentKey | String | Unique key of tournament |

Example:

```
{
    "data":[
        {
           "tournamentKey": "1855-2266-833649_P",
            "tournamentPlayerId": 0-456,
            "operatorToken": "1234",
            "playerName": " player1",
            "balanceAmount": 10000,
            "registerAllowed": 98,
            "reviveAllowed": 3,
            "status": 1
        },
        {
           "tournamentKey": "1865-2266-833649_P",
            "tournamentPlayerId": 0-457,
            "operatorToken": "1234",
            "playerName": " player1",
            "balanceAmount": 10000,
            "registerAllowed": 98,
            "reviveAllowed": 3,
            "status": 1
        },
        {
           "tournamentKey": "1866-2266-833649_P",
            "tournamentPlayerId": 0-458,
            "operatorToken": "1234",
            "playerName": " player2",
            "balanceAmount": 10000,
            "registerAllowed": 98,
            "reviveAllowed": 3,
            "status": 1
        }
      ],
    "error": null
}
```

### 2.4.5 Get Tournament Cash Prize Reward List

To get cash prize reward list of the tournament.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentAutoRewardCashPrize
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| start_time | Long | Yes | Time range of record's start times and end times<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 14 days** |
| end_time | Long | Yes | |
| Status | Integer | Yes | Reward status<br>**1: Receive reward successfully**<br>**0: Failed to receive reward** |
| transaction_id | Long | No | Unique identity of reward transaction |
| tournament_ids | Long | Yes | Unique identity of tournament |
| player_name | String | No | Unique identity of tournament players |
| currency | String | No | Currency used in tournament |
| page_number | Integer | No | Batch number of data return<br>**Default value: 1** |
| row_count | Integer | No | Number of records for each batch<br>**Default value: 5000** |

Example:

URL:

```
https://api.pg-bo.me/external/Tournament/v2/GetTournamentAutoRewardCashPrize?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&start_time=1559145500000&end_time=1559145600000&status=1
```

## Response

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of tournament records |
| totalPage | Integer | Total pages of tournament result |
| result | JSON Array | List of winners in tournament |

*result* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| tournamentId | Integer | Unique identity of tournament |
| currency | string | Currency used in the tournament |
| winners | JSON Array | List of winners in tournament |

*winners* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | Long | Unique identity of reward transaction |
| transactionTime | Long | Time of reward transaction<br>(Unix time stamp in milliseconds) |
| tournamentPlayerId | String | Unique identity of tournament player |
| playerName | String | Unique identity of player |
| rank | Integer | Player ranking of tournament |
| adjustmentAmount | Decimal | Reward amount of player in tournament |
| message | String | Error message will pop up if there is an error |

Example:

```
{
    "data": {
        "totalCount": 3,
        "result": [
            {
                "tournamentId": 175,
                "currency": "CNY",
                "winners": [
                    {
                        "transactionId": 27021004,
                        "transactionTime": 1599811942000,
                        "tournamentPlayerId": 0-456,
                        "playerName": "calven77407A5275",
                        "rank": 2,
                        "adjustmentAmount": 20000.00000,
                    }
                ]
            },
            {
                "tournamentId": 184,
                "currency": "CNY",
                "winners": [
                    {
                        "transactionId": 27021007,
                        "transactionTime": 1599811943000,
                        "tournamentPlayerId": 0-457,
                        "playerName": "JiaXiangHong",
                        "rank": 1,
                        "adjustmentAmount": 200.00000
                    },
                    {
                        "transactionId": 27021008,
                        "transactionTime": 1599811943000,
                        "tournamentPlayerId": 0-458,
                        "playerName": "dong",
                        "rank": 2,
                        "adjustmentAmount": 100.00000
                    }
                ]
            }
        ],
        "totalPage": 1
    },
    "error": null
}
```

## 2.4.6 Create Tournament

To create a tournament.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/CreateTournament
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_info | JSON Array | Yes | Tournament Info |
| has_sub_tournaments | Boolean | Yes | To identify if the tournament has Sub-Tournaments |
| game_mode | Uint | Yes | Point Tournament:<br>1: Accumulation<br>2: Depletion<br>3: Hybrid<br><br>Cash Tournament:<br>4: Profit Percentage<br>5: Win Percentage<br>6: Profit Amount<br>7: Win Amount<br>8: Win count<br>9: Spin Count<br>10: Total Bet |
| qualified_point | Decimal | Yes | Required points for ranks |
| initial_point | Decimal | Yes | Initial points for tournament (only for point tournament) |
| game_ids | Uint Array | Yes | Game for tournament |
| currency | String Array | Yes | Currency used in the tournament |
| start_time | Long | Yes | Tournament start time |
| end_time | Long | Yes | Tournament end time |
| operator_token | String | Yes | Unique identity of operator |

| | | | |
|---|---|---|---|
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| maximum_revive_allowed | Uint | No | Maximum number of times to re-play tournament |
| maximum_register_allowed | Uint | No | Maximum number of times to re-register tournament |
| sub_tournament_info | JSON Array | Depends on has_sub_tournament | Sub-Tournament information |
| number_of_player_limit | Uint | No | Maximum number of players to register tournament |
| minimum_bet_amount | Decimal | No | Minimum bet amount to register for the tournament |
| minimum_bet_count | Uint | No | Minimum number of bet count to register tournament |
| leaderboard_player | Uint | No | Number of players to show in the leaderboard |
| player_multiplier | Uint | No | Win multiplier for all players in the tournament |
| required_register | Bool | No | Allows players to manually register for current tournament |

*tournament_info* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| language | String | Tournament Language |
| info | JSON Array | Tournament Information |

*Info* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| name | String | Name of the Tournament |
| description | String | Additional description |
| rules | String | Rules for the Tournament |
| winning_criteria | String | Winning Requirements |
| registration_fees | String | Registration fees of Tournament |
| registration_url | String | Tournament registration URL |
| prize_type | Uint | Type of Tournament prize<br>0: Item prize<br>1: Cash prize<br>**Note:**<br>Cash prize will automatically be transferred to player's wallet after the tournament ends. |
| prize | JSON Array | Prize information |

*Prize* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| position_to | Integer | Range of the number of prize winners.<br>Example: 3 = the current prize would be for 1st to 3rd winners. |
| item | String | Prize Item<br>**Note:**<br>Cash prize will be given based on the currency base rate of the game. |

*sub_tournament_info* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| game_ids | Uint Array | Sub-Tournament Game IDs |
| start_time | Long | Sub-Tournament Start Time |
| end_time | Long | Sub-Tournament End Time |

Example:

URL:
```
https://api.pg-bo.me/external/Tournament/v2/CreateTournament?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
tournament_info={"language":"fi-
FI","info":{"name":"TournamentName","description":"TournamentDesc","rules":"Tourna
mentRules","winning_criteria":"WinningCriteria","registration_fees":"RegistrationF
ees","registration_url":"RegistrationUrl","prize_type":0,"prize":{"winner_prize_de
tails":[{"position_to":1,"item":"1st prize"},{"position_to":3,"item":"2rd
prize"},{"position_to":10,"item":"3rd
prize"}]}}}&sub_tournament_info={"game_ids":[1,2,3],"start_time":1613623305000,"en
d_time":1613796105000}&has_sub_tournaments=true&start_time=1613623305000&end_time=
1613796105000&currency=JPY&qualified_point=200&initial_point=199&game_mode=1&game_
ids=1&game_ids=2&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&maximum_revi
ve_allowed=10&maximum_register_allowed=10&number_of_player_limit=10&minimum_bet_am
ount=3.1&minimum_bet_count=5&leaderboard_player=100&player_multiplier=2&required_r
egister=true
```

## Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| tournamentId | Uint | Create Tournament ID |

Example:
```
{
    "data": {
        "tournamentId": 1859
    },
    "error": null
}
```

### 2.4.7 Update Tournament

To update a tournament.

**Request**
API URL : {PgSoftAPIDomain}/Tournament/v2/UpdateTournament
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Integer | Yes | Unique identity of tournament |
| start_time | Long | Yes | Tournament start time |
| end_time | Long | Yes | Tournament end time |
| qualified_point | Decimal | Yes | Required points for ranks |
| game_ids | Integer Array | Yes | Games for tournament |
| currency | String Array | Yes | List of Tournament currency |
| number_of_player_limit | Integer | No | Maximum number of players to register tournament |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |

```
Example:
URL:
https://api.pg-bo.me/external/Tournament/v2/UpdateTournament?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311

Body:
tournament_id=1757&start_time=1604016000000&end_time=1604102400000&qualified_point
=9639&game_ids=2&game_ids=1&currency=JPY&currency=CNY&number_of_player_limit=9639&
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl
```

## Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| data | Boolean | Update status |

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

## 2.4.8 Update Sub-Tournament Information

To update a Sub-Tournament's information.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/UpdateSubTournament
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Uint | Yes | Unique identity of tournament |
| sub_tournament_info | JSON Array | Yes | Sub-Tournament information |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| participant_tokens | String Array | Yes | Unique identity of operator |

*sub_tournament_info* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| sub_tournament_id | Uint | Unique identity of sub tournament |
| Interval_start_time | Long | Sub-Tournament Start / End Time (must be within start /end time of the Main Tournament) |
| Interval_end_time | Long | |
| game_ids | Uint Array | Sub-Tournament games |

```
Example:
URL:
https://api.pg-bo.me/external/Tournament/v2/UpdateSubTournament?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311

Body:
tournament_id=1859&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&sub_tournament_info:{"sub_tournament_id":1737,"interval_start_time":1604016000000,"interval_end_time":1604102400000,"game_ids":[1,2]}&participant_tokens=abc123&participant_tokens=abc124
```

## Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| data | Boolean | Update status |

Example:

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

### 2.4.9 Get Tournament Information

To get a tournament's information.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/GetTournamentInfoLanguageByTournamentId
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Uint | Yes | Unique identity of tournament |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |

```
Example:
URL:
https://api.pg-
bo.me/external/Tournament/v2/GetTournamentInfoLanguageByTournamentId?trace_id=b3f3
7e57-2873-40b1-aa95-f126c25ed311

Body:
tournament_id=1845&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl
```

**Response**

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| tournamentInfoId | Uint | Unique identity of tournament info id |
| language | String | Tournament supported language |
| name | String | Name of the Tournament |
| description | String | Additional description |
| rules | String | Rules for the Tournament |
| winningCriteria | String | Winning Requirements |
| registrationFees | String | Registration fees of Tournament |
| registrationUrl | String | Tournament registration Url |
| prizeType | Uint | Type of tournament prize<br>0: Item prize<br>1: Cash prize |

| | | |
|---|---|---|
| prize | JSON Array | Tournament winner prize details |
| createdTime | Long | Tournament created time |
| updatedTime | Long | Tournament updated time |
| createdBy | String | API or BackOffice user who registered the player into the Tournament |
| updatedBy | String | API or Backoffice user who updated the Tournament |

Example:

```
{
    "data": [
        {
            "tournamentInfoId": 1876,
            "language": "ja-JA",
            "name": "ILName",
            "description": "ILDesc",
            "rules": "ILRules",
            "winningCriteria": "ILWinningCt",
            "registrationFees": "ILRegistrationFees",
            "registrationUrl": "ILRegistrationUrl",
            "prizeType": 1,
            "prize": {
                "winnerPrizeDetails": {
                    "1": {
                        "positionFrom": 1,
                        "positionTo": 1,
                        "item": "myItem"
                    },
                    "2": {
                        "positionFrom": 2,
                        "positionTo": 2,
                        "item": "myItem2"
                    },
                    "3": {
                        "positionFrom": 3,
                        "positionTo": 3,
                        "item": "myItem3"
                    }
                }
            },
            "createdTime": 1606969557000,
            "updatedTime": 1607315386000,
            "createdBy": "External API",
            "updatedBy": "External API"
        }
    ],
    "error": null
}
```

## 2.4.10 Create Tournament Information

To create a tournament's information.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/CreateTournamentInfoLanguage
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Uint | Yes | Unique identity of tournament |
| name | String | Yes | Name of the Tournament |
| language | String | Yes | Tournament supported languages |
| description | String | No | Additional description |
| rules | String | No | Rules for the Tournament |
| winning_criteria | String | No | Winning Requirements |
| registration_fees | String | No | Registration fees of Tournament |
| registration_url | String | No | Tournament registration URL |
| prize | JSON Array | No | Tournament prize |
| prize_type | Uint | No | Type of tournament prize<br>0: Item prize<br>1: Cash prize |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| participant_tokens | String Array | Yes | Unique identity of operator |

*prize* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| winner_prize_details | JSON Array | Tournament winner's prize |

*prize* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| position_to | Integer | Range of the number of prize winners.<br>Example: 3 = the current prize would be for 1st to 3rd winners. |
| item | String | Prize Item |

```
Example:
URL:
https://api.pg-
bo.me/external/Tournament/v2/CreateTournamentInfoLanguage?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311

Body:
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&tournament_id=1845name=InfoLa
nguageName&language=ja-
JA&description=XXX&rules=XXX&winning_criteria=XXX&registration_fees=XXX&registrati
on_url=XXX&prize:{"winner_prize_details":[{"position_to":1,"item":"myItem"},{"posi
tion_to":2,"item":"myItem2"},{"position_to":3,"item":"myItem3"}]}&prize_type=1&par
ticipant_tokens=abc123&participant_tokens=abc123
```

## Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| tournamentInfoId | Uint | Unique identity of tournament info |

```
Example:

{
    "data": 1895,
    "error": null
}
```

## 2.4.11 Update Tournament Information

To update an existing tournament's information.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/UpdateTournamentInfoLanguage
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Uint | Yes | Unique identity of tournament |
| tournament_info_id | String | Yes | Unique identity of tournament info |
| name | String | Yes | Name of the Tournament |
| language | String | Yes | Tournament supported language |
| description | String | No | Additional description |
| rules | String | No | Rules for the Tournament |
| winning_criteria | String | No | Winning Requirements |
| registration_fees | String | No | Registration fees of Tournament |
| registration_url | String | No | Tournament registration Url |
| prize | JSON Array | No | Tournament prize |
| prize_type | Uint | No | Type of Tournament prize<br>0: Item prize<br>1: Cash prize |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| participant_tokens | String Array | Yes | Unique identity of operator |

*prize* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| winner_prize_details | JSON Array | Tournament winner's prize |

*prize* JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| position_to | Integer | Range of the number of prize winners.<br>Example: 3 = the current prize would be for 1st to 3rd winners. |
| item | String | Prize Item |

```
Example:
URL:
https://api.pg-
bo.me/external/Tournament/v2/UpdateTournamentInfoLanguage?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311

Body:
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&tournament_id=1845&tournament
_info_id=1876&name=ILName&language=ja-
JA&description=ILDesc&rules=ILRules&winning_criteria=ILWinningCt&registration_fees
=ILRegistrationFees&registration_url=ILRegistrationUrl&prize:{"winner_prize_detail
s":[{"position_to":1,"item":"myItem"},{"position_to":2,"item":"myItem2"},{"positio
n_to":3,"item":"myItem3"}]}&prize_type=1&participant_tokens=abc123&participant_tok
ens=abc123
```

## Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| data | Boolean | Update status |

```
Example:

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

## 2.4.12 Delete Tournament Information

To delete a specific information from a tournament.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/DeleteTournamentInfoLanguage
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_id | Uint | Yes | Unique identity of tournament |
| tournament_info_id | String | Yes | Unique identity of tournament info id |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |

```
Example:
URL:
https://api.pg-bo.me/external/Tournament/v2/DeleteTournamentInfoLanguage?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311

Body:
tournament_id=1845&tournament_info_id=1875&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl
```

**Response**

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| data | Boolean | Update status |

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

## 2.4.13 Remove a Tournament Player

To remove a Tournament player and their progress.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/CancelTournamentPlayer
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| tournament_player_id | String | Yes | Unique identity of tournament player id |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |

```
Example:
URL:
https://api.pg-
bo.me/external/Tournament/v2/CancelTournamentPlayer?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311

Body:
tournament_player_id=843662&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl
```

**Response**

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| data | Boolean | Remove player status |

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

## 2.4.14 Revive Players

To reset the tournament progress for a specific player.

**Request**

API URL : {PgSoftAPIDomain}/Tournament/v2/RevivePlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| player_name | String | Yes | Unique identity of player name |
| sub_tournament_id | Uint | Yes | Unique identity of Sub-Tournament |
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |

```
Example:
URL:
https://api.pg-bo.me/external/Tournament/v2/RevivePlayers?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311

Body:
tournament_player_id=843662&operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl
```

### Response

Body Parameters:

| Parameter name | Data type | Description |
|---|---|---|
| tournamentKey | String | Unique key of tournament |
| tournamentPlayerId | String | Unique identity of tournament player id |
| operatorToken | String | Unique identity of operator |
| playerName | String | Unique identity of player name |
| balanceAmount | Decimal | Amount given to player when revived |
| registerAllowed | Integer | Numbers of times a player is allowed to re-register for the tournament |
| reviveAllowed | Integer | Numbers of times a player is allowed to be revived in a tournament |

```
Example:
{
    "data": {
        "tournamentKey": "1870-2283-843667_P",
        "tournamentPlayerId": 0-456,
        "operatorToken": "a**d",
        "playerName": "hltestdev",
        "balanceAmount": 200.00000,
        "registerAllowed": 9,
        "reviveAllowed": 9
    },
    "error": null
}
```

## 2.5 Bet History [Latest version: v4]

PG provides APIs to get various types of bet history data. Please take note that API data will be only available within 3-5 minutes after each spin.
Only bets with completed status will be provided. Operator may check through PG Back Office for pending and failed bets.

**Note:**
External Bet History API is a dedicated API which using standalone URL: **{DataGrabAPIDomain}**

## 2.5.1 Get History

To get bet history.

### Notes

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.
- PG may re-push data by renewing the data row version. Operator is required to identify and update duplicated data by *betId.*
- Data row version cannot be a unique value. Operator is required to grab at least 1500 records for every request.

### Steps

- **Step 1:** In the very first GetHistory API call, set row_version = 1
- **Step 2:** Save the maximum rowVersion in each data grabbing request call
- **Step 3:** In the subsequence calls, set the row_version value to the saved rowVersion in Step 2.
- Repeat step 2 and 3 until records returned are lesser than the required count (e.g.: 1500 records per request)
- If the number of records returned are less than the required count, stop, and wait for a time interval (recommended 5 minutes) for the next API call
- Identify duplicate records by checking duplicate betId in each call

## Request

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |
| bet_type | Integer | Yes | Bet types of bet record:<br>1: Real game |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds)<br>**Note:**<br>• **Set value to 1 for first call**<br>• **Operator is required to save maximum row_version for each call, to be used as row_version value for next call request** |
| hands_status | Long | No | Status of hands:<br>0: All (Default)<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |

Example:

URL:
```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetHistory?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&row_ver
sion=1346592723000&hands_status=0
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| betId | Ulong | Unique identity of child bet (Unique key value) |
| parentBetId | Ulong | Unique identity of parent bet |
| playerName | String | Unique identity of player |
| currency | String | Currency used by player in the record |
| gameId | Integer | Unique identity of game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| betAmount | Decimal* | Turnover of player |
| winAmount | Decimal* | Pay-out of player |
| jackpotRtpContributionAmount | Decimal* | Jackpot RTP contribution amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| balanceBefore | Decimal* | Balance of player before transaction |
| balanceAfter | Decimal* | Balance of player after transaction |
| handsStatus | Integer | Status of hand:<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| betTime | Integer | Start time of current bet<br>(Unix time stamp in milliseconds) |
| betEndTime | Integer | End time of current bet<br>(Unix time stamp in milliseconds) |
| isFeatureBuy | Boolean | To indicate the spin type:<br>True: Feature spin<br>False: Normal spin |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
       {
                   "betId": 1443839146530971234,
                   "parentBetId": 1443839146530971234,
                   "playerName": "player1",
                   "currency": "CNY",
                   "gameId": 92,
                   "platform": 1,
                   "betType": 1,
                   "transactionType": 1,
                   "betAmount": 500.0,
                   "winAmount": 0.0,
                   "jackpotRtpContributionAmount": 0.0,
                   "jackpotContributionAmount": 0.0,
                   "jackpotWinAmount": 0.0,
                   "balanceBefore": 78962.39,
                   "balanceAfter": 78462.39,
                   "handsStatus": 2,
                   "rowVersion": 1633073283123,
                   "betTime": 1633073052123,
                   "betEndTime": 1633073052123,
                   "isFeatureBuy": false
               },
               {
                   "betId": 1443839117799981234,
                   "parentBetId": 1443839028721351234,
                   "playerName": "player1",
                   "currency": "CNY",
                   "gameId": 92,
                   "platform": 1,
                   "betType": 1,
                   "transactionType": 1,
                   "betAmount": 0.0,
                   "winAmount": 0.0,
                   "jackpotRtpContributionAmount": 0.0,
                   "jackpotContributionAmount": 0.0,
                   "jackpotWinAmount": 0.0,
                   "balanceBefore": 79462.39,
                   "balanceAfter": 79462.39,
                   "handsStatus": 2,
                   "rowVersion": 1633073284123,
                   "betTime": 1633073045123,
                   "betEndTime": 1633073045123,
                   "isFeatureBuy": false
               }
       ],
    "error": null
}
```

## 2.5.2 Get History for Specific Time

To get the bet history for a specific time range.

### Notes

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.
- We recommend operators delay grabbing the data after an hour as the data may update within an hour.
- We also recommend operator always grab the data through the automated job for the data an hour ago.
- Other than that, we strongly recommend operator to grab the data by Get History API instead of Get History for Specific Time API to prevent the data may update within an hour.

### Steps

- **Step 1:** In the very first GetHistory API call, set from_time = unix timestamp of your desired time, and set to_time as current time
- **Step 2:** Save the maximum betEndTime in each data grabbing request call
- **Step 3:** In the subsequence calls, set the from_time parameter value to the saved betEndTime in step 2, to_time parameter value may remain unchanged (current time)
- Repeat step 2 and 3 until records returned are lesser than the required count (e.g.: 1500 records per request)
- If number of records returned are less than the required count, stop, and wait for a time interval (recommended 5 minutes) for the next API call
- Identify duplicate records by checking duplicate betId in each call

### Request

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |
| bet_type | Integer | Yes | Bet types of bet record:<br>1: Real game |
| from_time | Long | Yes | Time range of the start time and end time of the bet records<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 40 days** |
| to_time | Long | Yes | |

Example:

URL:

```
https://api.pg-bo.me/external-
datagrabber/Bet/v4/GetHistoryForSpecificTimeRange?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&from_ti
me=1346592723000&to_time=1546592723000
```

**Response**

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| betId | Ulong | Unique identity of child bet (Unique key value) |
| parentBetId | Ulong | Unique identity of parent bet |
| playerName | String | Unique identity of player |
| currency | String | Currency used by player in the record |
| gameId | Integer | Unique identity of game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| betAmount | Decimal* | Turnover of player |
| winAmount | Decimal* | Pay-out of player |
| jackpotRtpContributionAmount | Decimal* | Jackpot RTP contribution amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| balanceBefore | Decimal* | Balance of player before transaction |
| balanceAfter | Decimal* | Balance of player after transaction |
| handsStatus | Integer | Status of hand:<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| betTime | Integer | Start time of current bet<br>(Unix time stamp in milliseconds) |
| betEndTime | Integer | End time of current bet<br>(Unix time stamp in milliseconds) |
| isFeatureBuy | Boolean | To indicate the spin type:<br>True: Feature spin<br>False: Normal spin |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
                {
                    "betId": 1443924025750851234,
                    "parentBetId": 1443924025750851234,
                    "playerName": "player1",
                    "currency": "CNY",
                    "gameId": 89,
                    "platform": 1,
                    "betType": 1,
                    "transactionType": 1,
                    "betAmount": 6.0,
                    "winAmount": 0.0,
                    "jackpotRtpContributionAmount": 0.0,
                    "jackpotContributionAmount": 0.0,
                    "jackpotWinAmount": 0.0,
                    "balanceBefore": 17762.94,
                    "balanceAfter": 17756.94,
                    "handsStatus": 2,
                    "rowVersion": 1633093531234,
                    "betTime": 1633093281234,
                    "betEndTime": 1633093281234,
                    "isFeatureBuy": false
                },
                {
                    "betId": 1443924030033241234,
                    "parentBetId": 1443924030033241234,
                    "playerName": "player1",
                    "currency": "CNY",
                    "gameId": 89,
                    "platform": 1,
                    "betType": 1,
                    "transactionType": 1,
                    "betAmount": 6.0,
                    "winAmount": 36.0,
                    "jackpotRtpContributionAmount": 0.0,
                    "jackpotContributionAmount": 0.0,
                    "jackpotWinAmount": 0.0,
                    "balanceBefore": 17756.94,
                    "balanceAfter": 17786.94,
                    "handsStatus": 1,
                    "rowVersion": 1633093521234,
                    "betTime": 1633093291234,
                    "betEndTime": 1633093291234,
                    "isFeatureBuy": false
                }
        ],
    "error": null
}
```

### 2.5.3 Get Single Player History

To get bet history of specific player.

#### Notes

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.

#### Request

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of player |
| bet_type | Integer | Yes | Bet types of bet record:<br>1: Real game |
| start_time | Long | Yes | Time range of start times and end times of bet records<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 7 days** |
| end_time | Long | Yes | |

Example:

URL:
```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetPlayerHistory?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&start_time=1346592723000&end_time=1546592723000
```

**Response**

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| betId | Ulong | Unique identity of child bet (Unique key value) |
| parentBetId | Ulong | Unique identity of parent bet |
| playerName | String | Unique identity of player |
| currency | String | Currency used by player in the record |
| gameId | Integer | Unique identity of game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| betAmount | Decimal* | Turnover of player |
| winAmount | Decimal* | Pay-out of player |
| jackpotRtpContributionAmount | Decimal* | Jackpot RTP contribution amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| balanceBefore | Decimal* | Balance of player before transaction |
| balanceAfter | Decimal* | Balance of player after transaction |
| handsStatus | Integer | Status of hand:<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| betTime | Integer | Date of current record<br>(Unix time stamp in milliseconds) |
| betEndTime | Integer | End time of current bet<br>(Unix time stamp in milliseconds) |
| isFeatureBuy | Boolean | To indicate the spin type:<br>True: Feature spin<br>False: Normal spin |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
              {
                  "betId": 1443458363362385920,
                  "parentBetId": 1443458363362385920,
                  "playerName": "fongying",
                  "currency": "CNY",
                  "gameId": 94,
                  "platform": 1,
                  "betType": 1,
                  "transactionType": 1,
                  "betAmount": 5.0,
                  "winAmount": 0.0,
                  "jackpotRtpContributionAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "jackpotWinAmount": 0.0,
                  "balanceBefore": 10001704.78,
                  "balanceAfter": 10001699.78,
                  "handsStatus": 2,
                  "rowVersion": 1632982509049,
                  "betTime": 1632982266626,
                  "betEndTime": 1632982266654,
                  "isFeatureBuy": false
              },
              {
                  "betId": 1443458366315175936,
                  "parentBetId": 1443458366315175936,
                  "playerName": "fongying",
                  "currency": "CNY",
                  "gameId": 94,
                  "platform": 1,
                  "betType": 1,
                  "transactionType": 1,
                  "betAmount": 5.0,
                  "winAmount": 0.0,
                  "jackpotRtpContributionAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "jackpotWinAmount": 0.0,
                  "balanceBefore": 10001699.78,
                  "balanceAfter": 10001694.78,
                  "handsStatus": 2,
                  "rowVersion": 1632982515184,
                  "betTime": 1632982267325,
                  "betEndTime": 1632982267354,
                  "isFeatureBuy": false
              }
       ],
    "error": null
}
```

### 2.5.4 Get Player's Daily Summary

To get player's daily bet history.

**Notes**

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.
- PG may re-push data by renewing data row version. Operator is required to identify, and update duplicated data by *betId.*
- Data row version cannot be a unique value. Operator is required to grab at least 1500 records for every request.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |
| bet_type | Integer | Yes | Category of bet record:<br>1: Real game |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds) |
| time_zone | Integer | No | Time zone:<br>-5: UTC-5<br>-4: UTC-4<br>0: UTC (Default)<br>1: UTC+1<br>4: UTC+4<br>7: UTC+7<br>8: UTC+8 |

Example:

URL:
```
https://api.pg-bo.me/external-
datagrabber/Bet/v4/GetPlayerDailySummary?trace_id=b3f37e57-2873-40b1-aa95-
f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&row_ver
sion=1346592723000&time_zone=8
```

### Response

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity of player |
| playerType | Integer | Category of player:<br>1: Real player<br>2: Test player |
| gameId | Integer | Unique identity of game |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| currency | String | Currency used by player in the record |
| hands | Integer | Counts of main bets |
| games | Integer | Counts of child bets |
| betAmount | Decimal* | Turnover of player |
| winLossAmount | Decimal* | Win loss amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| summaryDate | Integer | Date of current record<br>(Unix time stamp in milliseconds) |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "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
}
```

### 2.5.5 Get Player's Daily Summary for Specific Time

To get player daily bet history for specific time range.

#### Notes

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.
- PG may re-push data by renewing data row version. Operator is required to identify, and update duplicated data by *playerName*, *gameId* and *summaryDate.*

#### Request

API URL : {DataGrabAPIDomain}/Bet/v4/GetPlayerDailySummaryForSpecificTimeRange
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |
| bet_type | Integer | Yes | Category of bet record:<br>1: Real game |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds) |
| from_time | Long | Yes | Time range of start times and end times of bet records<br>(Unix time stamp in milliseconds)<br>**Note:**<br>• **Value range 1 day -40 days** |
| to_time | Long | Yes | |
| time_zone | Integer | No | Time zone:<br>-5: UTC-5<br>-4: UTC-4<br>0: UTC (Default)<br>1: UTC+1<br>4: UTC+4 |

| | | | |
|---|---|---|---|
| | | | 8: UTC+8 |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetPlayerDailySummaryForSpecificTimeRange?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&from_time=1346592723000&to_time=1546592723000&time_zone=8
```

### Response

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity of player |
| playerType | Integer | Category of player:<br>1: Real player<br>2: Test player |
| gameId | Integer | Unique identity of game |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| currency | String | Currency used by player in the record |
| hands | Integer | Counts of main bets |
| games | Integer | Counts of child bets |
| betAmount | Decimal* | Turnover of player |
| winLossAmount | Decimal* | Win loss amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| summaryDate | Integer | Date of current record<br>(Unix time stamp in milliseconds) |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "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
}
```

### 2.5.6 Get Hands Summary

To get the game summary records of each hand.

**Notes**

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.
- PG may re-push data by renewing data row version. Operator is required to identify, and update duplicated data by *betId.*
- Data row version cannot be a unique value. Operator is required to grab at least 1500 records for every request.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |
| bet_type | Integer | Yes | Bet types of bet record:<br>1: Real game |
| hands_status | Long | Yes | Status of hands:<br>0: All<br>1: Incomplete<br>2: Completed |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds) |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetHandsSummary?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
`operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&hands_status=2&row_version=1346592723000`

**Response**

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| parentBetId | Ulong | Unique identity of parent bet |
| playerName | String | Unique identity of player |
| gameId | Integer | Unique identity of game |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| currency | String | Currency used by player in the record |
| betAmount | Decimal* | Turnover of player |
| winAmount | Decimal* | Pay-out of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| balanceBefore | Decimal* | Balance of player before transaction |
| balanceAfter | Decimal* | Balance of player after transaction |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| handStartTime | Integer | Start time of current hand<br>(Unix time stamp in milliseconds) |
| handEndTime | Integer | End time of current hand<br>(Unix time stamp in milliseconds) |
| handsStatus | Integer | Status of hand:<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |
| collapseSpinCount | Integer | Number of normal spin collapsed |
| collapseFreeSpinCount | Integer | Number of free spin collapsed |
| isFeatureBuy | Boolean | To indicate the spin type:<br>True: Feature spin<br>False: Normal spin |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
              {
                  "parentBetId": 1443100640863068160,
                  "playerName": "fongying",
                  "gameId": 103,
                  "betType": 1,
                  "transactionType": 1,
                  "platform": 1,
                  "currency": "CNY",
                  "betAmount": 6.0,
                  "winAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "jackpotWinAmount": 0.0,
                  "balanceBefore": 10001632.88,
                  "balanceAfter": 10001626.88,
                  "rowVersion": 1632897431960,
                  "handStartTime": 1632896978967,
                  "handEndTime": 1632896978967,
                  "handStatus": 2,
                  "collapseSpinCount": 0,
                  "collapseFreeSpinCount": 0,
                  "isFeatureBuy": false
              },
              {
                  "parentBetId": 1443100653622140928,
                  "playerName": "fongying",
                  "gameId": 103,
                  "betType": 1,
                  "transactionType": 1,
                  "platform": 1,
                  "currency": "CNY",
                  "betAmount": 6.0,
                  "winAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "jackpotWinAmount": 0.0,
                  "balanceBefore": 10001626.88,
                  "balanceAfter": 10001620.88,
                  "rowVersion": 1632897431960,
                  "handStartTime": 1632896982003,
                  "handEndTime": 1632896982003,
                  "handStatus": 2,
                  "collapseSpinCount": 0,
                  "collapseFreeSpinCount": 0,
                  "isFeatureBuy": false
              }
       ],
    "error": null
}
```

## 2.5.7 Get Hourly Hands Summary

To get operator's hourly hands summary records.

### Notes

- Operator can retrieve the bet history for the 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.

### Request

API URL : {DataGrabAPIDomain}/Bet/v4/GetHandsSummaryHourly
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| from_time | Long | Yes | Time range of start times and end times of bet records<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 40 days** |
| to_time | Long | Yes | |
| currency | String | No | Currency used by player in the record |
| transaction_types | Integer Array | No | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |

Example:

URL:

```
https://api.pg-bo.me/external-
datagrabber/Bet/v4/GetHandsSummaryHourly?trace_id=b3f37e57-2873-40b1-aa95-
f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&from_time=1567296000000&to_ti
me=1569801600000&currency=cny&transaction_types=1&transaction_types=2
```

**Response**

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| dateTime | Integer | Date and time of a record per hour |
| totalHands | Integer | Total count of game hands |
| currency | String | Currency used by player in the record |
| totalBetAmount | Decimal* | Total amount of turnover |
| totalWinAmount | Decimal* | Total amount of pay-out |
| totalPlayerWinLossAmount | Decimal* | Total amount of player’s win and loss |
| totalCompanyWinLossAmount | Decimal* | Total amount of company’s win and loss |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| totalCollapseSpinCount | Integer | Total count of normal spin collapsed |
| totalCollapseFreeSpinCount | Integer | Total count of free spin collapsed |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data": [
        {
            "dateTime": 1633093200123,
            "totalHands": 6,
            "currency": "CNY",
            "totalBetAmount": 23.8,
            "totalWinAmount": 114.0,
            "totalPlayerWinLossAmount": 90.2,
            "totalCompanyWinLossAmount": -90.2,
            "transactionType": 1,
            "totalCollapseSpinCount": 8,
            "totalCollapseFreeSpinCount": 0
        },
        {
            "dateTime": 1633093200123,
            "totalHands": 13,
            "currency": "EUR",
            "totalBetAmount": 15.0,
            "totalWinAmount": 3.4,
            "totalPlayerWinLossAmount": -11.6,
            "totalCompanyWinLossAmount": 11.6,
            "transactionType": 1,
            "totalCollapseSpinCount": 2,
            "totalCollapseFreeSpinCount": 0
        },
        {
            "dateTime": 1633093200213,
            "totalHands": 2,
            "currency": "THB",
            "totalBetAmount": 80.0,
            "totalWinAmount": 0.0,
            "totalPlayerWinLossAmount": -80.0,
            "totalCompanyWinLossAmount": 80.0,
            "transactionType": 1,
            "totalCollapseSpinCount": 0,
            "totalCollapseFreeSpinCount": 0
        }
      ],
    "error": null
}
```

## 2.5.8 Get Summary of Total Player Bets

To get the summary of total player bets for specific time range.

### Notes

- Operator can retrieve the bet history for the last 40 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.

### Request

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| bet_type | Integer | Yes | Bet types of bet record:<br>1: Real game |
| from_time | Long | Yes | Time range of start times and end times of bet records<br>(Unix time stamp in milliseconds)<br>**Note:**<br>**Value range 1 day - 40 days.**<br>**The time will be rounded off hourly.**<br>**Example:**<br>**If the request sent by operator is 1616250600000 (20/3/2021 2:30:00 PM) ~ 1616254200000 (20/3/2021 3:30:00 PM), system will return the data from 1616252400000 (20/3/2021 3:00:00 PM) to 1616256000000 (20/3/2021 4:00:00 PM)** |
| to_time | Long | Yes | |

| | | | |
|---|---|---|---|
| currency | String | No | Currency used by player in the record |
| game_ids | Integer Array | No | Unique identity of games |
| player_modes | Integer Array | No | Mode of game:<br>0: Normal<br>1: Cash bonus |
| player_type | Integer | No | Type of player:<br>0: All<br>1: Real player<br>2: Test player |
| transaction_type | Integer | No | Type of transaction:<br>0: All<br>1: Cash<br>2: Bonus<br>3: Free Game |
| time_zone | Integer | No | Time zone:<br>-5: UTC-5<br>-4: UTC-4<br>0: UTC (Default)<br>1: UTC+1<br>4: UTC+4<br>8: UTC+8 |

Example:

`URL:`

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetPlayerBetsSummationTotal?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

`Body:`

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&bet_type=1&from_time=1346592723000&to_time=1346592723000
```

**Response**

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| totalCollapseSpinCount | Integer | Total count of normal spin collapsed |
| totalCollapseFreeSpinCount | Integer | Total count of free spin collapsed |
| currency | String | Currency used by player in the record |
| totalRowCount | Integer | Total count of game rows |
| totalGames | Integer | Total count of game rounds |
| totalHands | Integer | Total count of game hands |
| totalBetAmountConverted | Decimal* | Total bet amount of the player converted (currency as EUR) |
| totalWinLossAmountConverted | Decimal* | Total win loss amount of the player converted (currency as EUR) |
| totalCompanyWinLossAmountConverted | Decimal* | Total win loss amount of the company converted (currency as EUR) |
| totalJackpotRtpContributionAmountConverted | Decimal* | Total jackpot RTP contribution amount of the records converted (currency as EUR) |
| totalJackpotContributionAmountConverted | Decimal* | Total jackpot contribution amount of the records converted (currency as EUR) |
| totalJackpotWinAmountConverted | Decimal* | Total jackpot win amount of the records converted (currency as EUR) |
| totalBetAmount | Decimal* | Total bet amount of records |
| totalWinLossAmount | Decimal* | Total win loss amount of records |
| totalCompanyWinLossAmount | Decimal* | Total company win loss amount of records |
| totalJackpotRtpContributionAmount | Decimal* | Total jackpot RTP contribution amount of records |
| totalJackpotContributionAmount | Decimal* | Total jackpot contribution amount of records |
| totalJackpotWinAmount | Decimal* | Total jackpot win amount of records |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
                {
                    "totalCollapseSpinCount": 8,
                    "totalCollapseFreeSpinCount": 0,
                    "currency": "CNY",
                    "totalRowCount": 1,
                    "totalGames": 14,
                    "totalHands": 6,
                    "totalBetAmountConverted": 3.05092,
                    "totalWinLossAmountConverted": 11.56274,
                    "totalCompanyWinLossAmountConverted": -11.56274,
                    "totalJackpotRtpContributionAmountConverted": 0.0,
                    "totalJackpotContributionAmountConverted": 0.0,
                    "totalJackpotWinAmountConverted": 0.0,
                    "totalBetAmount": 23.8,
                    "totalWinLossAmount": 90.2,
                    "totalCompanyWinLossAmount": -90.2,
                    "totalJackpotRtpContributionAmount": 0.0,
                    "totalJackpotContributionAmount": 0.0,
                    "totalJackpotWinAmount": 0.0
                },
                {
                    "totalCollapseSpinCount": 2,
                    "totalCollapseFreeSpinCount": 0,
                    "currency": "EUR",
                    "totalRowCount": 1,
                    "totalGames": 16,
                    "totalHands": 13,
                    "totalBetAmountConverted": 15.0,
                    "totalWinLossAmountConverted": -11.6,
                    "totalCompanyWinLossAmountConverted": 11.6,
                    "totalJackpotRtpContributionAmountConverted": 0.0,
                    "totalJackpotContributionAmountConverted": 0.0,
                    "totalJackpotWinAmountConverted": 0.0,
                    "totalBetAmount": 15.0,
                    "totalWinLossAmount": -11.6,
                    "totalCompanyWinLossAmount": 11.6,
                    "totalJackpotRtpContributionAmount": 0.0,
                    "totalJackpotContributionAmount": 0.0,
                    "totalJackpotWinAmount": 0.0
                }
        ],
    "error": null
}
```

## 2.5.9 Get Player's Unfinished Games

To get a player unfinished hand.

### Notes

- Operator can retrieve the bet history for the last 60 days.
- This API is for reporting purposes only. We do not recommend using this API for any other business operations. Please contact PG if you require it to integrate with business operations.

### Request

| | |
|---|---|
| API URL | : {DataGrabAPIDomain}/Bet/v4/GetPlayerUnfinishedGamesBet |
| HTTP method | : POST |

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity for player |
| from_time | Long | Yes | Time range of tournaments' start times and end times<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 7 days** |
| to_time | Long | Yes | |
| game_ids | Integer Array | No | Unique identity of games |
| time_zone | Integer | No | Time zone:<br>-5: UTC-5<br>-4: UTC-4<br>0: UTC (Default)<br>1: UTC+1<br>4: UTC+4<br>8: UTC+8 |

Example:

<u>URL:</u>

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetPlayerUnfinishedGamesBet?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

<u>Body:</u>

```
operator_token=abcd&secret_key=abc123&game_ids=1&player_name=player123&date_from=1588723200000&date_to=1588982400000
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| parentBetId | String | Unique Identity of parent bet |
| gameId | Integer | Unique Identity of game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| platform | Integer | Platform of bet record |
| currency | String | Currency of record |
| betAmount | Decimal | Turnover of player |
| winAmount | Decimal | Pay-out of player |
| jackpotRtpContributionAmount | Decimal* | Jackpot RTP contribution amount of player |
| jackpotContributionAmount | Decimal | Jackpot contribution amount of player |
| balanceBefore | Decimal | Balance of player before transaction |
| balanceAfter | Decimal | Balance of player after transaction |
| handStartTime | Long | Start time of the hand<br>(Unix time stamp in milliseconds) |
| handEndTime | Long | End time of the hand<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "data":[
              {
                  "parentBetId": 1444964157178265600,
                  "gameId": 108,
                  "transactionType": 1,
                  "platform": 1,
                  "currency": "CNY",
                  "betAmount": 375.0,
                  "winAmount": 0.0,
                  "jackpotRtpContributionAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "balanceBefore": 10001446.89,
                  "balanceAfter": 10001071.89,
                  "handStartTime": 1633341275871,
                  "handEndTime": 1633341278912
              }
      ],
    "error": null
}
```

## 2.5.10 Get Bet Details

To retrieve the bet details of multiple bets

### Notes

- Operator can retrieve the bet history for the last 60 days.

### Request

API URL : {DataGrabAPIDomain}/Bet/v4/GetBetDetails
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| bet_ids | Ulong | Yes | Unique identity of bets |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetBetDetails?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abcd&secret_key=abc123&bet_ids=1315486539764989952&bet_ids=13154865
39764989960
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| betId | Ulong | Unique identity of child bet (Unique key value) |
| parentBetId | Ulong | Unique identity of parent bet |
| playerName | String | Unique identity of player |
| currency | String | Currency used by player in the record |
| gameId | Integer | Unique identity of game |
| platform | Integer | Platform of bet record<br>(Please refer to Platform for more information) |
| betType | Integer | Category of bet record:<br>1: Real game |
| transactionType | Integer | Category of transaction:<br>1: Cash<br>2: Bonus<br>3: Free game |
| betAmount | Decimal* | Turnover of player |
| winAmount | Decimal* | Pay-out of player |
| jackpotRtpContributionAmount | Decimal* | Jackpot RTP contribution amount of player |
| jackpotContributionAmount | Decimal* | Jackpot contribution amount of player |
| jackpotWinAmount | Decimal* | Jackpot win amount of player |
| balanceBefore | Decimal* | Balance of player before transaction |
| balanceAfter | Decimal* | Balance of player after transaction |
| handsStatus | Integer | Status of hand:<br>1: Non-last hand<br>2: Last hand<br>3: Adjusted |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| betTime | Integer | Start time of current bet<br>(Unix time stamp in milliseconds) |
| betEndTime | Integer | End time of current bet<br>(Unix time stamp in milliseconds) |
| isFeatureBuy | Boolean | To indicate the spin type:<br>True: Feature spin<br>False: Normal spin |

* System might return results up to 6 decimal places depending on game and balance

Example:

```
{
    "data":[
              {
                  "betId": 1443924025750851584,
                  "parentBetId": 1443924025750851584,
                  "playerName": "jj02",
                  "currency": "CNY",
                  "gameId": 89,
                  "platform": 1,
                  "betType": 1,
                  "transactionType": 1,
                  "betAmount": 6.0,
                  "winAmount": 0.0,
                  "jackpotRtpContributionAmount": 0.0,
                  "jackpotContributionAmount": 0.0,
                  "jackpotWinAmount": 0.0,
                  "balanceBefore": 17762.94,
                  "balanceAfter": 17756.94,
                  "handsStatus": 2,
                  "rowVersion": 1633093533869,
                  "betTime": 1633093289187,
                  "betEndTime": 1633093289215,
                  "isFeatureBuy": false
              }
       ],
    "error": null
}
```

## 2.5.11 Get Failed Bets

To retrieve failed bets.

Notes

- **Operator can retrieve the bet history for the last 60 days.**

**Request**

API URL : {DataGrabAPIDomain}/Bet/v4/GetFailedBets
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| from_time | Long | Yes | Time range of the start time and end times of enquiries<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 40 days** |
| to_time | Long | Yes | |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetFailedBets?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&from_time=1346592723000&to_time=1546592723000
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| **betId** | Integer | Unique identity of child bet (Unique key value) |
| **playerName** | String | Unique identity of player |
| **message** | String | Failed reason |
| **betStartTime** | Integer | Start time of current bet<br>(Unix time stamp in milliseconds) |

Example:

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

### 2.5.12 Pending Bets

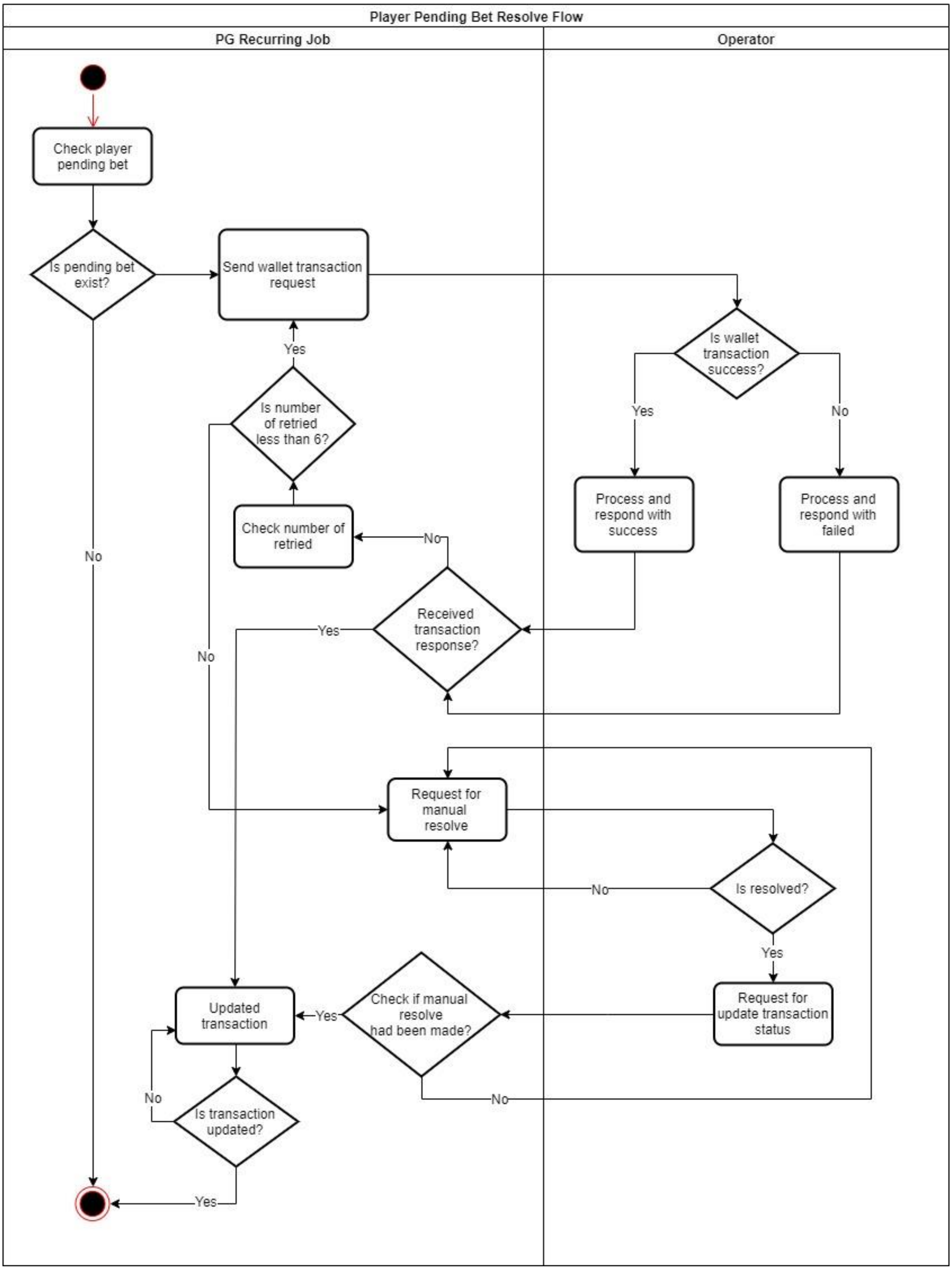
Sometimes pending bets may occur due to unforeseen circumstances during gameplay for example, network latency, system failure, system timeout and so on.

Generally, pending bet will be resolved automatically when the player relaunches the game. However, the bet will remain in pending status for some rare scenarios:

- Player did not relaunch the game after encountering an error
- PG Soft system did not receive any response from the operator when resolving pending bets
- PG Soft system receive an error response from the operator when resolving pending bets

Default interval of pending bet resolving time is set to 60 minutes. Pending bets which exceed the interval will be resolved by PG Soft system automatically. This interval is customizable, and every operator can have their own bet resolve interval time.

Manual resolve is required when failure to resolve pending bets after several attempts. In such case, operator is required to resolve the pending bet manually, and PG Soft will mark these bets' statuses as complete after getting the confirmation from the operator.


Player Pending Bet Resolve Flow
PG Recurring Job
Operator
Check player pending bet
Is pending bet exist?
Send wallet transaction request
Is wallet transaction success?
Yes
No
Process and respond with success
Process and respond with failed
Yes
Is number of retried less than 6?
Check number of retried
No
Received transaction response?
Yes
No
No
Request for manual resolve
Is resolved?
No
Yes
Request for update transaction status
Check if manual resolve had been made?
Yes
No
Updated transaction
Is transaction updated?
No
Yes

## 2.5.13 Get Pending Bets

To retrieve pending bets.

Notes

- **Operator can retrieve the bet history for the last 60 days.**

**Request**

API URL : {DataGrabAPIDomain}/Bet/v4/GetPendingBets
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_names | string | No | Unique identity of player |
| from_time | Long | Yes | Time range of the start time and end times of enquiries<br>(Unix time stamp in milliseconds)<br>**Note: Value range 1 day - 40 days** |
| to_time | Long | Yes | |
| count | Integer | Yes | Number of records for each batch<br>**Note: Value range 1500-5000** |

Example:

URL:
```
https://api.pg-bo.me/external-datagrabber/Bet/v4/GetPendingBets?trace_id=b3f37e57-
2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_names=testplayer1&play
er_names=testplayer2&count=5000&from_time=1346592723000&to_time=1546592723000
```

### Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity of player |
| gameId | String | Unique identity of game |
| betStatus | Integer | Status of bet<br>1: Pending |
| betId | Ulong | Unique identity of child bet (Unique key value) |
| betAmount | Decimal | Bet amount of game |
| betTime | Integer | Start time of current bet<br>(Unix time stamp in milliseconds) |

Example:

```
{
    "data":[
              {
                  "playerName": "player1",
                  "gameId": 38,
                  "betStatus": 1,
                  "betId": 1444876729369241234,
                  "betAmount": 500.0,
                  "betTime": 1633320431123
              }
       ],
    "error": null
}
```

### 2.5.14 Manual Resolve of Pending Bet

To manually resolve pending bet.

**Request**

API URL : {DataGrabAPIDomain}/Bet/v4/ManualResolveBet
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | string | Yes | Unique identity of player |
| bet_id | Integer | Yes | Unique identity of child bet (Unique key value) |
| is_success_bet | Boolean | Yes | Status of bet:<br>True: Successful bet<br>False: Failed bet |
| game_id | Integer | Yes | Unique identity of game |
| balance_before_adjustment | Decimal | No | Player balance before adjustment (For successful bet only) |

Example:

URL:

```
https://api.pg-bo.me/external-
datagrabber/Bet/v4/ManualResolveBet?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=testplayer1&bet_i
d=1325714346441543680&is_success_bet=true&game_id=1
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| resolveStatus | Boolean | Bet resolve result:<br>True: Successful<br>False: Failed |
| betStatus | Integer | Status of bet<br>0: Non-pending<br>1: Pending<br>2: Failed<br>3: Successful<br>4: Adjusted |

**Please refer to the following use cases for further details of API response:**

| Actual Bet Status | Is_Success_Bet Value in 1st Request | Is_Success_Bet Value in 2nd Request | API Response |
|---|---|---|---|
| Successful Bet | True | - | resolveStatus: False, betStatus: 0 |
| Successful Bet | True | True | resolveStatus: False, betStatus: 0 |
| Successful Bet | False | - | resolveStatus: False, betStatus: 0 |
| Successful Bet | False | False | resolveStatus: False, betStatus: 0 |
| Failed Bet | True | - | resolveStatus: False, betStatus: 0 |
| Failed Bet | False | - | resolveStatus: False, betStatus: 0 |
| Pending Bet | True | - | resolveStatus: True, betStatus: 3 |
| Pending Bet | True | True | resolveStatus: True, betStatus: 3 |
| Pending Bet | False | - | resolveStatus: True, betStatus: 2 |
| Pending Bet | False | False | resolveStatus: True, betStatus: 2 |
| Pending Bet | True | False | resolveStatus: False, betStatus: 3 |
| Pending Bet | False | True | resolveStatus: False, betStatus: 2 |

Example:

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

### Error Response

| Code | Message |
|---|---|
| 1307 | Invalid player game session |

## 2.5.15 Manual Reset of Pending Game State

To reset pending game state.

**Request**

API URL : {DataGrabAPIDomain}/Bet/v4/ManualResetGameState
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | string | Yes | Unique identity of player |
| game_id | Integer | Yes | Unique identity of game |
| operator_player_session | string | Yes | Operator session for player |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber/Bet/v4/ManualResetGameState?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=testplayer1&game_id=1
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| resetStatus | Boolean | Bet resolve result:<br>True: Successful<br>False: Failed |

Example:

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

## 2.6 Game [Latest version: v2]

### 2.6.1 Get Games List

To get list of games.

**Request**

API URL : {PgSoftAPIDomain}/Game/v2/Get
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | Yes | Currency of game's legal bet amount |
| language | String | No | Language of data content:<br>en-us: English<br>zh-cn: Chinese |
| status | Integer | No | Status of games:<br>0: Inactive games<br>1: Active games<br>**Note: If no value passed in, it would select for both inactive games and active games** |

Example:

URL:

```
https://api.pg-bo.me/external/Game/v2/Get?trace_id=b3f37e57-2873-40b1-aa95-
f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&language=en-
us&status=1
```

**Response**

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| gameId | Integer | Unique identity of game |
| gameName | String | Given name of game |
| gameCode | String | Unique code of game |
| gameLegalBetAmounts | JSON Array | Legal bet amount of games |
| status | Integer | Global status of game:<br>0: Inactive<br>1: Active<br>2: Suspended<br>**Note: Game will only be accessible if game status and game release status are active** |
| releaseStatus | Integer | Status of game for operator:<br>0: Inactive<br>1: Active<br>2: Suspended<br>**Note: Game will only be accessible if game status and game release status are active** |
| isSupportFreeGame | Boolean | To indicate if free game is supported:<br>True: Supported<br>False: Not supported |
| **category** | Integer | Category of game:<br>1: Slot game<br>2: Card game |

gameLegalBetAmounts JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| gameId | Integer | Unique identity of game |
| gameTypeId | Integer | Type of game (Card game only):<br>0: None<br>1: Baccarat-Supersix<br>2: Baccarat-Commission |
| legalBetAmounts | JSON Array | Available legal bet amount of games |

legalBetAmounts JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| coinSize | Decimal | Bet size of game |
| betMultiplier | Integer | Bet level of game |
| baseBet | Integer | Base bet of game |
| betAmount | Decimal | Bet amount of game |

Example:

```
{
    "data”: [
        {
            "gameId": 1,
            "gameName": "Honey Trap of Diao Chan",
            "gameCode": "diaochan",
            "status": 1,
            "releaseStatus": 1,
            "isSupportFreeGame": true,
            "category": 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
                        }
                    ]
                }]
        }
```

## 2.6.2 Get Game Legal Bet Amounts

To get the legal bet amounts of the games.

**Request**

API URL : {PgSoftAPIDomain}/Game/v2/GetGameLegalBetAmounts
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | Yes | Currency of game's legal bet amount |
| game_ids | Integer Array | Yes | Unique identity of games |

Example:

URL:
```
https://api.pg-bo.me/external/Game/v2/GetGameLegalBetAmounts?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&currency=CNY&game_ids=1&game_ids=2
```

### Response

Body Parameters:

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| gameId | Integer | Unique identity of game |
| gameTypeId | Integer | Type of game (Card game only):<br>0: None<br>1: Baccarat-Supersix<br>2: Baccarat-Commission |
| legalBetAmounts | JSON Array | Available legal bet amount of games |

legalBetAmounts JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| coinSize | Decimal | Bet size of game |
| betMultiplier | Integer | Bet level of game |
| baseBet | Integer | Base bet of game |
| betAmount | Decimal | Bet amount of game |

Example:

```
{
    "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
}
```

### 2.6.3 Change Operator’s Game Release Status

To change the game release status on operator’s side.

**Request**

API URL : {PgSoftAPIDomain}/Game/v2/UpdateGameStatus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity for operator |
| secret_key | String | Yes | Shared passphrase between PG Soft and operator |
| release_status | Integer | Yes | Release status of game:<br>0: Inactive<br>1: Active |
| game_ids | Integer Array | Yes | Unique identity of games |

Example:

URL:
```
https://api.pg-bo.me/external/Game/v2/UpdateGameStatus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&release_status=1&game_ids=1&game_ids=2
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| actionResult | Boolean | Result of action request:<br>true: Successful<br>false: Failed |

Example:

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

## 2.6.4 Get Game Cert

To get game certificates.

**Request**

API URL : {PgSoftAPIDomain}/Game/v2/GetGamesCert
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| game_ids | Integer Array | No | Unique id of each game<br>**Note: If no value passed in, it would select for all games** |
| game_codes | String Array | No | Unique code of each game<br>**Note: If no value passed in, it would select for all games** |

Example:

URL:
```
https://api.pg-bo.me/external/Game/v2/GetGamesCert?trace_id=b3f37e57-2873-40b1-
aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&game_ids=1&game_code=diaochan
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| game_id | Integer | Unique id of each game |
| game_name | String | Unique name of each game |
| game_code | String | Unique code of each game |
| cert | JSON Array | List of the certificates |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| hash | String | Hash value of the game file |
| file | String | Name of the game file |

Example:

```
{
    "data": [
        {
            "game_id": 1,
            "game_name": "Honey Trap of Diao Chan",
            "game_code": "diaochan",
            "cert": [
                {
                    "hash": "e2cac0a05c7407c32fd50c4e6fb6779161dfeecc",
                    "file": "PG.Common.GameCore.RNG.dll"
                },
                {
                    "hash": "c4280d2afa00fd17e6b037bd7f4a1fa53b620775",
                    "file": "PG.Common.GameCore.Slot.Engine.dll"
                },
                {
                    "hash": "144c068a7011bb22e57e967e3561200ae2b8f41e",
                    "file": "PG.Game.DiaoChan.Engine.dll"
                },
                {
                    "hash": "bb95b970f2b57c62c96c111e8ddb6ac3961b0881",
                    "file": "PG.SlotGames.Source.DiaoChan.dll"
                }
            ]
        }
    ],
    "error": null
}
```

## 2.7 Cash [Latest version: v2]

### 2.7.1 Get Wallet Transaction

To get the wallet transaction.

**Note**

• Operator can only retrieve record for the last 40 days.

**Request**

API URL : {DataGrabAPIDomain}/cash/v2/GetWalletTransactions
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds) |
| player_name | String | No | Unique identity of player |
| row_count | Integer | No | Number of records for each batch<br>**Note:**<br>• **Default value: 1**<br>• **Value range: 1-5000** |
| currency | String | No | Currency of player |

Example:

URL:

```
https://api.pg-bo.me/external-
datagrabber/cash/v1/GetWalletTransactions?trace_id=b3f37e57-2873-40b1-aa95-
f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&row_version=1346592723000&pla
yer_name=testplayer1&row_count=10&currency=CNY
```

### Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| lastRowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| result | JSON Array | List of the result |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| transactionId | String | Unique identity of transaction |
| playerName | String | Unique identity of player |
| currencyCode | String | Currency of player |
| transactionType | Integer | Transaction type:<br>101: Payout<br>201: Bet |
| transactionAmount | Decimal | Transaction amount |
| transactionFrom | Decimal | Balance before transaction |
| transactionTo | Decimal | Balance after transaction |
| transactionDateTime | Long | Transaction date and time |

Example:

```
{
    "data": {
        "lastRowVersion": 1598617560000,
        "result": [
            {
                "transactionId": 13777918,
                "playerName": "Player1",
                "currencyCode": "CNY",
                "transactionType": 100,
                "transactionAmount": 2.00,
                "transactionFrom": 20001873.53,
                "transactionTo": 20001875.53,
                "transactionDateTime": 1598617560000
            }
        ]
    },
    "error": null
}
```

### 2.7.2 Get Manual Adjustment Details

To get the manual adjustment details.

**Note**
• Operator can only retrieve record for the last 60 days.

**Request**
API URL : {DataGrabAPIDomain}/Cash/v2/GetManualAdjustmentDetails
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| currency | String | No | Currency of player |
| row_version | Long | Yes | Updated time of data<br>(Unix time stamp in milliseconds) |
| row_count | Integer | No | Number of records for each batch<br>**Note:**<br>• **Default value: 1**<br>• **Value range: 1-5000** |
| transaction_type | String | No | Transaction type:<br>900: External Adjustment<br>901: Tournament Adjustment |
| transaction_reference_id | String | No | Reference Id for adjustment |
| status | Integer | No | Status of manual adjustment<br>1: Success<br>0: Pending |

Example:

URL:

```
https://api.pg-bo.me/external-datagrabber
/Cash/v2/GetManualAdjustmentDetails?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&row_count=10&currency=CNY&row
_version=1346592723000&row_count=1&transaction_type=901&transaction_reference_id=a
djustment-001&status=1
```

## Response

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| lastRowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| result | JSON Array | List of the result |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity of player |
| operatorToken | String | Unique identity of operator |
| currencyCode | String | Currency of player |
| transactionReferenceId | String | Reference Id for adjustment |
| transactionType | Integer | Transaction type:<br>900: External Adjustment<br>901: Tournament Adjustment |
| transactionAmount | Decimal | Adjustment amount |
| transactionDateTime | Long | Adjustment date and time |
| **status** | Integer | Status of manual adjustment<br>1: Success<br>0: Pending |
| **transactionRemark** | String | Remark for Adjustment |

Example:

```
{
    "data": {
        "lastRowVersion": 1624444862653,
        "result": [
            {
                "playerName": "abc",
                "operatorToken": "aabbcc",
                "currencyCode": "CNY",
                "transactionReferenceId": "CASH-1234-1234567",
                "transactionType": 901,
                "transactionAmount": 1000.0,
                "transactionDateTime": 1624444862653,
                "status": 1,
                "transactionRemark": ""
            }
        ]
    },
    "error": null
}
```

## 2.8 Player [Latest version: v3]

### 2.8.1 Get Online Player Count

To get the total number of online players.

**Request**

API URL : {PgSoftAPIDomain}/Player/v3/GetOnlinePlayerCount
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity for operator |
| secret_key | String | Yes | Shared passphrase between PG Soft and operator |
| game_id | Integer | No | Unique identity for game<br>**Note: System will return list of online players for all games if parameter value is not provided** |

Example:

URL:
```
https://api.pg-bo.me/external/Player/v3/GetOnlinePlayerCount?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&game_id=1
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| totalCount | Integer | Total number of records |
| totalPlayerCount | Integer | Total number of online players |
| onlinePlayerSummary | JSON Array | List of online players for games |

JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds) |
| gameId | Integer | Unique identity for game |
| playerCount | Integer | Total number of online players |
| status | Integer | Status of player<br>1: Online |

Example:

```
{
    "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
}
```

### 2.8.2 Get Players' Online Status

To get the online statuses of the last 10 players.

**Request**

API URL : {PgSoftAPIDomain}/Player/v3/GetPlayersOnlineStatus
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity for operator |
| secret_key | String | Yes | Shared passphrase between PG Soft and operator |
| player_names | String Array | Yes | Unique identity of player |

Example:

URL:
```
https://api.pg-bo.me/external/Player/v3/GetPlayersOnlineStatus?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_names=player1&player_names=player2
```

## Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity for player |
| gameId | String | Unique identity for game |
| status | Integer | Online status of player<br>0: Offline<br>1: Online |

Example:

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

### 2.8.3 Get Online Player List

To get a list of online players.

**Request**

API URL : {PgSoftAPIDomain}/Player/v3/GetOnlinePlayers
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity for operator |
| secret_key | String | Yes | Shared passphrase between PG Soft and operator |
| **row_count** | Integer | No | Number of records for each batch<br>**Note:**<br>• **Default value: 1000**<br>• **Value range: 1-1000** |
| **row_version** | Long | No | Updated time of data<br>(Unix time stamp in milliseconds)<br>**Default value: 1** |

Example:

URL:

```
https://api.pg-bo.me/external/Player/v3/GetOnlinePlayers?trace_id=b3f37e57-2873-
40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&row_count=10
```

## Response

Body Parameters:
JSON Array

| Parameter name | Data type | Description |
|---|---|---|
| playerName | String | Unique identity for player |
| gameId | Integer | Unique identity for game |
| rowVersion | Long | Updated time of data<br>(Unix time stamp in milliseconds)<br>**Default value: 1** |

Example:

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

### 2.8.4 Create Player

To create a new player’s account in PG game system.

**Note**
Implementation of this API is optional as player account will be created automatically for a first-time player to access PG games.

**Request**

| | |
|---|---|
| API URL | : {PgSoftAPIDomain}/v3/Player/Create |
| HTTP method | : POST |

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of players<br>• player name is not case sensitive<br>• Only alphabet, number, and “@”, “-”, ”_” symbols are allowed<br>**Note: Max 50 characters** |
| nickname | String | No | Nickname of player<br>**Note: Max 50characters** |
| currency | String | Yes | Currency of player |

Example:

URL:
```
https://api.pg-bo.me/external/v3/Player/Create?trace_id=b3f37e57-2873-40b1-aa95-
f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&currenc
y=EUR&nickname=player123
```

## Response

Body Parameters:
JSON Object

| Parameter name | Data type | Description |
|---|---|---|
| action_result | Integer | Status of the request<br>1: successful<br>0: failed |

Example:

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

## Error Response

| Code | Message |
|---|---|
| 1034 | Invalid request |
| 1035 | Operation Failed |
| 1200 | Internal server error |
| 1204 | Invalid operator |
| 1305 | Invalid player (player already existed) |
| 1315 | Player has another operation is in progress |

## 2.8.5 Kick Out Player

To kick out a player from a PG game. The player will still be able to re-login to the game.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of players |

Example:

URL:
```
https://api.pg-bo.me/external/v3/Player/Kick?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data Type | Description |
|---|---|---|
| action_result | Integer | Status of the request<br>1: successful<br>0: failed |

Example:

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

### Error Response

| Code | Message |
|---|---|
| 1034 | Invalid request |
| 1035 | Operation Failed |
| 1200 | Internal server error |
| 1204 | Invalid operator |
| 1305 | Invalid player |
| 1315 | Player has another operation is in progress |

### 2.8.6 Suspend Player

To suspend a player's account. The suspended players will not be able to access PG games anymore.

**Request**

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>• **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of players |
| reason | String | Yes | Reason to suspend the player |

Example:

URL:
```
https://api.pg-bo.me/external/v3/Player/Suspend?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&reason=suspendedplayer
```

### Response

Body Parameters:
JSON Object

| Parameter name | Data Type | Description |
|---|---|---|
| action_result | Integer | Status of the request<br>1: successful<br>0: failed |

Example:

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

### Error Response

| Code | Message |
|---|---|
| 1034 | Invalid request |
| 1035 | Operation Failed |
| 1200 | Internal server error |
| 1204 | Invalid operator |
| 1305 | Invalid player |
| 1315 | Player has another operation is in progress |

### 2.8.7 Resume Player

To resume a suspended player's account.

**Request**

API URL : {PgSoftAPIDomain}/v3/Player/Reinstate
HTTP method : POST

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of players |
| reason | String | Yes | Reason of resume suspended player |

Example:

URL:

```
https://api.pg-bo.me/external/v3/Player/Reinstate?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&reason=resumesuspendedplayer
```

## Response

Body Parameters:
JSON Object

| Parameter name | Data Type | Description |
|---|---|---|
| action_result | Integer | Status of the request<br>1: successful<br>0: failed |

Example:

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

## Error Response

| Code | Message |
|---|---|
| 1034 | Invalid request |
| 1035 | Operation Failed |
| 1200 | Internal server error |
| 1204 | Invalid operator |
| 1305 | Invalid player |
| 1315 | Player has another operation is in progress |

## 2.8.8 Check Player Status

To check a specific player's account status.

### Note

This API is not to check an online player's status, it is just to check the player's account status in PG system. To check for an online active player, please check using the back office.

### Request

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

URL Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| trace_id | String | Yes | Unique identity (GUID) of the request<br>**Note:**<br>- **Please set the parameter value as GUID format** |

Body Parameters:

| Parameter name | Data Type | Mandatory | Description |
|---|---|---|---|
| operator_token | String | Yes | Unique identity of operator |
| secret_key | String | Yes | Shared passphrase between PGSoft and operator |
| player_name | String | Yes | Unique identity of players |

Example:

URL:
```
https://api.pg-bo.me/external/v3/Player/Check?trace_id=b3f37e57-2873-40b1-aa95-f126c25ed311
```

Body:
```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123
```

**Response**

Body Parameters:

JSON Object

| Parameter name | Data Type | Description |
|---|---|---|
| player_name | String | Unique identity of players |
| status | Integer | Status of the request<br>0: Disabled<br>1: Active<br>2: Inactive/Deactivated<br>3: Suspended<br>4: Player has another operation is in progress |

Example:

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

# 3. Appendix

## 3.1.1 Platform

| Code | Category | Platform |
|---|---|---|
| 1 | Web | Windows |
| 2 | | macOS |
| 3 | | Android |
| 4 | | iOS |
| 5 | | others |
| 6 | Cordova | Android |
| 7 | | iOS |
| 8 | Electron | Windows |
| 9 | | macOS |
| 10 | Native | Windows |
| 11 | | macOS |
| 12 | | Android |
| 13 | | iOS |
| 98 | others | System |
| 99 | | Smartbot |