

PG SOFT

Integration Document

External API v1.8

Contents

Revision History.....	5
1. Overview	10
2. API Methods	11
2.1 Free Game [Latest version: v1]	12
2.1.1 Get Free Games List	12
2.1.2 Get List of Free Game Converted to Bonus/Cash	16
2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash	18
2.1.4 Get Free Game Players List	20
2.1.5 Get List of Unregistered Players in Free Game	23
2.1.6 Get All Player Details for Free Game.....	26
2.1.7 Get Single Player Details in Free Game.....	29
2.1.8 Create Free Game	33
2.1.9 Create Free Game by Bet Amount	35
2.1.10 Transfer Player to Standard Free Game.....	37
2.1.11 Transfer Player to Flexible Free Game	39
2.1.12 Cancel Free Game	42
2.1.13 Cancel Player Free Game	43
2.1.14 Cancel Free Game of Unregistered Player	44
2.1.15 Cancel All Free Games of a Player.....	45
2.2 Bonus [Latest version: v1]	46
2.2.1 Get Bonus Game List.....	46
2.2.2 Get List of Bonus Games Converted to Cash	49
2.2.3 Get Bonus Game Players List	51
2.2.4 Get List of Unregistered Players of Bonus Game	54
2.2.5 Get All Player Details in Bonus Game	57
2.2.6 Get Single Player Details in Bonus Game	60
2.2.7 Create Bonus Game	63
2.2.8 Transfer Player to Standard Bonus Game.....	65
2.2.9 Transfer Player to Flexible Bonus Game	67
2.2.10 Cancel Bonus Game	70
2.2.11 Cancel Bonus Games of Registered Player	71
2.2.12 Cancel Bonus Game of Unregistered Player	72
2.2.13 Cancel All Bonus Games of a Player.....	73
2.3 Tournament [Latest version: v2]	74

2.3.1	Get Tournaments List.....	74
2.3.2	Get Tournament Players List.....	78
2.3.3	Get Tournament Top Rankings	81
2.3.4	Register Tournament Players.....	83
2.3.5	Get Tournament Cash Prize Reward List.....	85
2.4	Bet History [Latest version: v4]	88
2.4.1	Get History	89
2.4.2	Get History for Specific Time	93
2.4.3	Get Single Player History.....	97
2.4.4	Get Player's Daily Summary	100
2.4.5	Get Player's Daily Summary for Specific Time	103
2.4.6	Get Hands Summary	106
2.4.7	Get Hourly Hands Summary.....	109
2.4.8	Get Summary of Total Player Bets	111
2.4.9	Get Player's Unfinished Games.....	114
2.4.10	Get Bet Details	117
2.4.11	Get Failed Bets	120
2.4.12	Pending Bets	122
2.4.12.1	Get Pending Bets.....	124
2.4.12.2	Manual Resolve Pending Bet	126
2.4.13	Manual Reset Pending Game State	128
2.5	Game [Latest version: v2]	129
2.5.1	Get Games List	129
2.5.2	Get Game Legal Bet Amounts.....	132
2.5.3	Change Operator's Game Release Status	134
2.6	Jackpot [Latest version: v1]	135
2.6.1	Get Jackpots List.....	135
2.7	Player [Latest version: v3]	137
2.7.1	Get Online Player Count.....	137
2.7.2	Get Players Online Status.....	139
2.7.3	Get Online Players.....	140
2.7.4	Create Player.....	142
2.7.5	Kick Out Player	144
2.7.6	Suspend Player	145
2.7.7	Resume Player.....	146

2.7.8	Check Player Status	147
3.	Appendix	148
3.1	Platform	148

Revision History

Version	Date	Description
1.0	2019-04-03	First draft
1.1	2019-04-12	<p>Added new content</p> <ul style="list-style-type: none"> 2.1.5 Get Free Game Unregistered Players 2.1.12 Cancel All Free Games of Player 2.2.4 Get Bonus Unregistered Players 2.2.11 Cancel All Bonus of Player 2.4.6 Get Hands Summary <p>Amended Overview</p> <p>Amended API request parameter</p> <ul style="list-style-type: none"> 2.1.6 Create Free Game 2.1.11 Cancel Unregistered Player Free Game 2.2.5 Create Bonus 2.2.10 Cancel Unregistered Player Bonus <p>Amended API description</p> <ul style="list-style-type: none"> 2 API Methods 2.1.2 Get List of Free Game Converted to Bonus 2.1.3 Get List of Free Game's Bonus Converted to Cash 2.1.10 Cancel Registered Player Free Game 2.1.11 Cancel Unregistered Player Free Game 2.2.2 Get List of Bonus Converted to Cash 2.2.9 Cancel Registered Player Bonus 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	<p>Added new contents</p> <ul style="list-style-type: none"> 2.1.6 Get Free Game All Players 2.1.7 Get Player Free Games 2.2.5 Get Bonus All Players 2.2.6 Get Player Bonus <p>Amended API descriptions & request descriptions</p> <ul style="list-style-type: none"> 2.1.4 Get Free Game Players 2.1.5 Get Free Game Unregistered Players 2.2.3 Get Bonus Players 2.1.5 Get Bonus Unregistered Players 2.4.6 Get Hands Summary 2.5.1 Get Game List <p>Amended API request</p> <ul style="list-style-type: none"> 2.1.5 Get Free Game Unregistered Players

		<ul style="list-style-type: none"> 2.1.7 Create Free Game 2.2.4 Get Bonus Unregistered Players 2.2.6 Create Bonus Amended API response <ul style="list-style-type: none"> 2.1.6 Get Free Game All Players 2.1.7 Create Free Game 2.2.5 Get Bonus All Players 2.2.6 Create Bonus 2.4.6 Get Hands Summary 2.5.2 Get Game List
1.4	2019-07-12	Added new contents <ul style="list-style-type: none"> 2.1.9 Create Free Game by Bet Amount 2.3.1 Get Tournaments 2.3.2 Get Tournament Players 2.5.2 Get Game Legal Bet Amount Amended API <ul style="list-style-type: none"> 2.5.1 Get Game List (version 2) Amended API request <ul style="list-style-type: none"> 2.1.8 Create Free Game Amended API response <ul style="list-style-type: none"> 2.1.1 Get Free Games
1.5	2019-08-22	Amended API description <ul style="list-style-type: none"> 2.1.1 Get Free Games 2.1.2 Get List of Free Game Converted to Bonus/Cash 2.1.3 Get List of Free Game's Bonus Converted to Cash 2.1.4 Get Free Game Players 2.1.6 Get Free Game All Players 2.2.1 Get Bonus 2.2.2 Get List of Bonus Converted to Cash 2.2.3 Get Bonus Players 2.2.5 Get Bonus All Players 2.3.1 Get Tournaments 2.3.2 Get Tournament Players 2.4.5 Get Player Daily Summary for Specific Time Amend API request <ul style="list-style-type: none"> 2.1.7 Get Player Free Games 2.2.9 Transfer Player to Flexible Bonus 2.4.1 Get History Amended API response <ul style="list-style-type: none"> 2.4.1 Get History 2.4.2 Get History for Specific Time

		<ul style="list-style-type: none"> 2.4.3 Get Single Player History 2.4.6 Get Hands Summary Added new contents <ul style="list-style-type: none"> 2.4.7 Get Player Bets Summation Total
1.6	2019-11-13	Added new contents <ul style="list-style-type: none"> 2.4.7 Get Hourly Hands Summary Amended API version <ul style="list-style-type: none"> 2.4 Bet History Amended API description <ul style="list-style-type: none"> 2.4 Bet History 2.4.1 Get History Amended API response <ul style="list-style-type: none"> 2.4.1 Get History 2.4.2 Get History for Specific Time 2.4.3 Get Single Player History
1.7	2020-02-10	Added new contents <ul style="list-style-type: none"> 2.5.3 Change Game Release Status 2.7.1 Get Online Player Count 2.7.2 Get Player Online Status Amended API response <ul style="list-style-type: none"> 2.4.1 Get History 2.4.2 Get History for Specific Time 2.4.3 Get Single Player History 2.4.4 Get Player Daily Summary 2.4.5 Get Player Daily Summary for Specific Time 2.4.6 Get Hands Summary 2.4.7 Get Hourly Hands Summary 2.4.8 Get Player Bets Summation Total 2.6.1 Get Jackpot List Amended API response parameter description <ul style="list-style-type: none"> 2.1.6 Get Free Game All Players 2.1.7 Get Player Free Games 2.2.5 Get All Bonus Players 2.2.6 Get Player Bonus
1.8	2020-12-11	Added new API version <ul style="list-style-type: none"> 2.7 Player Added new API <ul style="list-style-type: none"> 2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash 2.4.13 Pending Bets 2.4.14 Manual Reset Pending Game State 2.7.3 Get Online Players

- 2.7.4 Create Player
- 2.7.5 Kick Out Player
- 2.7.6 Suspend Player
- 2.7.7 Resume Player
- 2.7.8 Check Player Status

Amended API title or description

- 2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash
- 2.1.10 Transfer Player to Standard Free Game
- 2.1.13 Cancel Player Free Game
- 2.2.6 Get Single Player Details in Bonus Game
- 2.2.7 Create Bonus Game
- 2.4 Bet History
- 2.4.1 Get History
- 2.4.2 Get History for Specific Time
- 2.4.3 Get Single Player History
- 2.4.4 Get Player's Daily Summary
- 2.4.5 Get Player's Daily Summary for Specific Time
- 2.4.6 Get Hands Summary
- 2.4.7 Get Hourly Hands Summary
- 2.4.8 Get Summary of Total Player Bets
- 2.4.9 Get Player's Unfinished Games
- 2.4.10 Get All Unsuccessful Bets
- 2.4.11 Get Bet Details
- 2.4.12 Get Failed Bets
- 2.6.1 Get Jackpots List
- 2.7.2 Get Players Online Status

Amended API request or response contents

- 2.1.1 Get Free Games List
- 2.1.4 Get Free Game Players List
- 2.1.5 Get List of Unregistered Players in Free Game
- 2.1.6 Get All Player Details for Free Game
- 2.1.7 Get Single Player Details in Free Game
- 2.1.8 Create Free Game
- 2.1.9 Create Free Game by Bet Amount
- 2.1.10 Transfer Player to Standard Free Game
- 2.1.11 Transfer Player to Flexible Free Game
- 2.2.1 Get Bonus Game List
- 2.2.2 Get List of Bonus Games Converted to Cash
- 2.2.3 Get Bonus Game Players List
- 2.2.4 Get List of Unregistered Players of Bonus Game
- 2.2.5 Get All Player Details in Bonus Game
- 2.2.6 Get Single Player Details in Bonus Game
- 2.2.9 Transfer Player to Flexible Bonus Game
- 2.2.13 Cancel All Bonus Games of a Player
- 2.3.1 Get Tournaments List

		<ul style="list-style-type: none"> • 2.3.2 Get Tournament Players List • 2.3.4 Register Tournament Players • 2.3.5 Get Tournament Cash Prize Reward List • 2.4.1 Get History • 2.4.2 Get History for Specific Time • 2.4.3 Get Single Player History • 2.4.4 Get Player's Daily Summary • 2.4.5 Get Player's Daily Summary for Specific Time • 2.4.6 Get Hands Summary • 2.4.7 Get Hourly Hands Summary • 2.4.8 Get Summary of Total Player Bets • 2.4.9 Get Player's Unfinished Games • 2.4.10 Get All Unsuccessful Bets • 2.4.11 Get Bet Details • 2.4.12 Get Failed Bets • 2.6.1 Get Jackpots List • 2.7.2 Get Players Online Status
--	--	--

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

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`

Response

API returns JSON-formatted responses. For success and failed API request, system will 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:

Success 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 Free Game [Latest version: v1]

2.1.1 Get Free Games List

To get the list of created free games.

Request

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

HTTP method : POST

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
from_time	Long	No	Time range of free games' start time (Unix time stamp in milliseconds)
to_time	Long	No	Time range of free games' end time (Unix time stamp in milliseconds)
from_created_time	Long	No	Time range of free games' created time (Unix time stamp in milliseconds)
to_created_time	Long	No	Time range of free games' created time (Unix time stamp in milliseconds)
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 0: Standard 3: Flexible
conversion_type	Char	No	Type of conversion after free game is completed: C: Cash B: Bonus
status	Integer	No	Status of free game: 0: Inactive 1: Active 2: Expired

Example:

```
operator_token=abc123&secret_key=a1b25cde5f3gh46ijk1&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: True: Flexible 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 amount of free game to cash conversion
maximumConversionAmount	Decimal	The maximum amount of free game to cash conversion
conversionType	Char	Type of conversion after free game is completed: C: Cash B: Bonus
bonusRatio	Decimal	Wagering requirement for bonus game. Refer to the multiple of the turnover in Free Game that need to be completed after the amount won in free game has transferred to bonus wallet
bonusRatioAmount	Decimal	Wagering requirement for bonus game.

		Refer to the total amount that need to be completed after the amount won in free game has transferred to the bonus wallet
bonusMinimumConversionAmount	Decimal	The maximum amount of bonus game to cash conversion
bonusMaximumConversionAmount	Decimal	The minimum amount of bonus game to cash conversion
bonusType	Integer	Bonus game conversion type: 0: Bonus 1: Cash 2: Bonus & Cash
status	Integer	Status of free game: 0: Inactive (Cancelled) 1: Active 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 (Unix time stamp in milliseconds)
createdDate	Long	Created date of free game (Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who created free game
updatedBy	String	API or BackOffice user who updated free game
isSupressDiscard	Boolean	Permission to disallow cancel offer True: Not allow player to cancel offer False: Allow 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
}

```

2.1.2 Get List of Free Game Converted to Bonus/Cash

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

Request

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

HTTP method : POST

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
free_game_id	Integer	Yes	Unique identity of free game
player_name	String	No	Unique identity of player Note: Fuzzy matching is implemented
from_transaction_time	Long	No	Time range of free game player's transaction (Unix time stamp in milliseconds)
to_transaction_time	Long	No	Time range of free game player's transaction (Unix time stamp in milliseconds)

Example:

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

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
  }
}
```

2.1.3 Get List of Free Game Converted to Bonus Game and Converted to Cash

To get list of games that was 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

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
free_game_id	Integer	Yes	Unique identity of free game
player_name	String	No	Unique identity of player Note: Fuzzy matching is implemented
from_transaction_time	Long	No	Time range of free game player's transaction (Unix time stamp in milliseconds)
to_transaction_time	Long	No	Time range of free game player's transaction (Unix time stamp in milliseconds)
status	Integer	No	Status of bonus game: 0: Inactive 1: Active 2: Expired

Example:

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 (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
}
```

2.1.4 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

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
free_game_id	Integer	Yes	Unique identity of free game
player_name	String	No	Unique identity of free game player Note: Fuzzy matching is implemented
from_time	Long	No	Time range of free game's start time
to_time	Long	No	(Unix time stamp in milliseconds)
from_created_time	Long	No	Time range of player registered into free game
to_created_time	Long	No	(Unix time stamp in milliseconds)
free_game_name	String	No	Given name of free game
status	Integer	No	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

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=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: G: Free Game B: Bonus
balanceAmount	Decimal	Player balance of free game
bonusRatioAmount	Decimal	Wagering requirement for bonus game. Refer to the total amount that need to be completed after the amount won in free game has transferred to the bonus wallet
bonusMaximumConversion Amount	Decimal	Maximum conversion amount for free game's bonus
freeGameMaximumConversion Amount	Decimal	Maximum conversion amount for free game
status	Integer	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
expiredDate	Long	Expiry date of free game (Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who created free game
updatedBy	String	API or BackOffice user who updated free game
updatedAt	Long	Date when player is registered or re-registered into free game (Unix time stamp in milliseconds)

createdDate	Long	Date when player is registered into free game (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,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "freeGameCount": 6,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdBy": "operator",
        "updatedBy": "operator",
        "createdDate": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

2.1.5 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

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
free_game_id	Integer	Yes	Unique identity of free game
player_name	String	No	Unique identity of free game 's unregistered player Note: Fuzzy matching is implemented
created_by	String	No	API or BackOffice user who registered player
status	Integer	No	Status of wallet of unregistered player: 0: Inactive (Cancelled) 1: Pending 2: Transferred

Example:

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 games player pending wallet
playerName	String	Unique identity of free games un-registered player
status	Integer	Status of unregistered player's wallet: 0: Inactive (Cancelled) 1: Pending 2: Transferred
updatedAt	Long	Date of player registered or re-registered into free game (Unix time stamp in milliseconds)
createdAt	Long	Date of player registered into free game (Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who registered player
updatedAt	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,
        "updatedAt": 1554351552000,
        "createdBy": "External API",
        "updatedBy": "External API"
      },
      {
        "pendingWalletId": 140393,
        "playerName": "eddywang99999",
        "status": 1,
        "createdDate": 1554351682000,
        "updatedAt": 1554351682000,
        "createdBy": "External API",
        "updatedBy": "External API"
      }
    ]
  },
  "error": null
}
```

2.1.6 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

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
free_game_id	Integer	Yes	Unique identity of free game
player_name	String	No	Unique identity of free game's unregistered player Note: Fuzzy matching is implemented
from_time	Long	No	Time range of free game's start time (Unix time stamp in milliseconds)
to_time	Long	No	Time range of free game's end time (Unix time stamp in milliseconds)
from_created_time	Long	No	Time range of free game's created time (Unix time stamp in milliseconds)
to_created_time	Long	No	Time range of free game's created time (Unix time stamp in milliseconds)
free_game_name	String	No	Given name of free game
status	Integer	No	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

```
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: G: Free Game B: Bonus
balanceAmount	Decimal	Player balance of unconverted free game, or converted amount of converted free game
bonusRatioAmount	Decimal	Wagering requirement for bonus game. Refer to the total amount that need to be completed after the amount won in free game has transferred to the bonus wallet
bonusMinimumConversion Amount	Decimal	Minimum conversion amount for free game's bonus
bonusMaximumConversion Amount	Decimal	Maximum conversion amount for free game's bonus
freeGameMaximumConversion Amount	Decimal	Maximum conversion amount for free game
status	Integer	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
expiredDate	Long	Expiry date of free game (Unix time stamp in milliseconds)
updatedAt	Long	Date of player registered or re-registered into free game

		(Unix time stamp in milliseconds)
createdDate	Long	Date of player registered into free game (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,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "freeGameMaximumConversionAmount": 0,
        "freeGameCount": 6,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdDate": 1546918715000,
        "updatedAt": 1547028195000,
      }
    ]
  },
  "error": null
}
```

2.1.7 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

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
player_name	String	Yes	Unique identity of free game's unregistered player
status	Integer Array	No	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_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 (Unix time stamp in milliseconds)
createdDate	Long	Created date of free game (Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who created free game
updatedBy	String	API or BackOffice user who updated free game
isSupressDiscard	Boolean	Permission to disallow cancel offer True: Not allow player to cancel offer False: Allow player to cancel offer
status	Integer	Status of free game wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
conversionType	Char	Type of conversion after free game is completed: C: Cash 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
}
```


2.1.8 Create Free Game

To create a new free game.

Request

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

HTTP method : POST

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 (Unix time stamp in milliseconds)
bonus_minimum_conversion_amount	Decimal	No*	Minimum conversion amount for bonus NULL: No limit of conversion amount
bonus_maximum_conversion_amount	Decimal	No*	Maximum conversion amount for bonus NULL: unlimited conversion amount
minimum_conversion_amount	Decimal	No	Minimum conversion amount for free game NULL: no limit of conversion amount
maximum_conversion_amount	Decimal	No	Maximum conversion amount for free game NULL: unlimited conversion amount
bonus_type	Integer	No*	Bonus wagering condition 0: Bonus 1: Cash 2: Bonus & Cash
bonus_ratio_amount**	Decimal	No*	Wagering requirement for bonus game. Refer to the amount that need to be completed after the amount won in free game has transferred to bonus wallet
bonus_ratio**	Decimal	No*	Wagering requirement for bonus game. Refer to the multiple of the turnover in Free Game that need to be completed after the amount won in free game has transferred to bonus wallet Turnover = bonus ratio x converted bonus amount

conversion_type	Char	Yes	Type of conversion after free game is completed: C: Cash 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 (Note: Only games with same base bet are accepted)
is_event	Boolean	Yes	Type of free game: True: Flexible False: Standard
player_name	String	No	Unique identity of player. System will create free game and assign to this player directly
transaction_id	String	Yes	Unique identity of transaction. Note: System will return correspond free game id for duplicate transaction

*Only applicable for bonus conversion type

**Turnover condition of *bonus ratio* will be overridden if *bonus ratio amount* is provided

Example:

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

Response

Body Parameters:

JSON Object

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

Example:

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

2.1.9 Create Free Game by Bet Amount

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

Request

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

HTTP method : POST

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 (Unix time stamp in milliseconds)
bonus_minimum_conversion_amount	Decimal	No*	Minimum conversion amount for bonus NULL: no limit conversion amount
bonus_maximum_conversion_amount	Decimal	No*	Maximum conversion amount for bonus NULL: unlimited conversion amount
minimum_conversion_amount	Decimal	No	Minimum conversion amount for free game NULL: No limit conversion amount
maximum_conversion_amount	Decimal	No	Maximum conversion amount for free game NULL: unlimited conversion amount
bonus_type	Integer	No*	Bonus wagering condition 0: Bonus 1: Cash 2: Bonus & Cash
bonus_ratio_amount**	Decimal	No*	Wagering requirement for bonus game. Refer to the amount that need to be completed after the amount won in free game has transferred to bonus wallet
bonus_ratio**	Decimal	No*	Wagering requirement for bonus game. Refer to the multiple of the turnover in Free Game that need to be completed after the amount won in free game has transferred to bonus wallet Turnover = bonus ratio x converted bonus amount

conversion_type	Char	Yes	Type of conversion for after free game is completed C: Cash B: Bonus
bet_amount	Decimal	Yes	Desired bet amount for each game. 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 (Note: Only games with same base bet are accepted)
is_event	Boolean	Yes	Type of free game: True: Flexible False: Standard
player_name	String	No	Unique identity of player. System will create free game and assign to this player directly
transaction_id	String	Yes	Unique identity of transaction. Note: System will return correspond free game ID for duplicate transaction

*Only applicable for bonus conversion type

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

Example:

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

Response

Body Parameters:

JSON Object

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

Example:

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

2.1.10 Transfer Player to Standard Free Game

To add players into a created standard free game.

Request

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

HTTP method : POST

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:

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: 0: Failed 1: Success 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
}
```

2.1.11 Transfer Player to Flexible Free Game

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

Request

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

HTTP method : POST

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 has 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	<p>Is allow re-register for player</p> <p>True: No. of times a player can re-register in a free game</p> <p>False: Player is only allowed to register once in a free game.</p>

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 Value range: Value \geq 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 player Value range: Value \geq 1 Default value: 0 Note: System will use default conversion amount if value = 0
bonus_maximum_conversion_amount	Decimal	No*	Bonus maximum conversion amount that can be overridden for player Value range: Value \geq 1 Default value: 0 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 player Value range: Value \geq 1 Default value: 0 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 player Value range: Value \geq 1 Default value: 0 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:

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

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: 0: Failed 1: Success 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",
      "freeGameCount": 10,
      "status": 1
    }
  ],
  "error": null
}
```

2.1.12 Cancel Free Game

To cancel a created free game.

Request

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

HTTP method : POST

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&free_game_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.1.13 Cancel Player Free Game

To cancel a free game of a specific player.

Request

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

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&balance_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.1.14 Cancel Free Game of Unregistered Player

To cancel a free game for a specific unregistered player.

Request

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

HTTP method : POST

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&pending_wallet_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.1.15 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

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:

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

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.2 Bonus [Latest version: v1]

2.2.1 Get Bonus Game List

To get the created bonus games list.

Request

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

HTTP method : POST

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
from_time	Long	No	Time range of bonus game's start time (Unix time stamp in milliseconds)
to_time	Long	No	
from_created_time	Long	No	Time range of bonus game's created time (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: 0: Standard 3: Flexible
status	Integer	No	Status of bonus game: 0: Inactive 1: Active 2: Expired

Example:

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¤cy=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: True: Flexible False: Standard
gameIds	Integer Array	Unique identity of games
currencyCode	String	Currency of bonus game
bonusRatio	Decimal	Wagering requirement for bonus game. Refer to the multiple of the turnover that need to be completed
bonusRatioAmount	Decimal	Wagering requirement for bonus game. Refer to the total amount that need to be completed
bonusMinimumConversion Amount	Decimal	The maximum amount of free game to cash conversion
bonusMaximumConversion Amount	Decimal	The minimum amount of free game to cash conversion
bonusType	Integer	Bonus game wagering condition: 0: Bonus 1: Cash 2: Bonus & Cash
status	Integer	Status of bonus game: 0: Inactive (Cancelled) 1: Active 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 (Unix time stamp in milliseconds)
createdDate	Long	Created date of bonus game (Unix time stamp in milliseconds)
isSuppressDiscard	Boolean	Permission to disallow cancel offer True: Not allow player to cancel offer False: Allow 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
}
```


2.2.2 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

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
bonus_id	Integer	Yes	Unique identity of bonus
player_name	String	No	Unique identity of player Note: Fuzzy matching is implemented
from_transaction_time	Long	No	Time range of bonus player's transaction
to_transaction_time	Long	No	(Unix time stamp in milliseconds)

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&bonus_id=12345&player_name=player1&from_transaction_time=1346592723000&to_transaction_time=1346592723000

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 (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
}
```

}

2.2.3 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

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
bonus_id	Integer	Yes	Unique identity of bonus
player_name	String	No	Unique identity of bonus player Note: Fuzzy matching is implemented
from_time	Long	No	Time range of bonus game's start time
to_time	Long	No	(Unix time stamp in milliseconds)
from_created_time	Long	No	Time range of player registered into
to_created_time	Long	No	bonus game
			(Unix time stamp in milliseconds)
status	Integer	No	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

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: G: Free Game B: Bonus
balanceAmount	Decimal	Player balance of bonus
bonusRatioAmount	Decimal	Wagering requirement for bonus game. Refer to the total amount that need to be completed
bonusMinimumConversion Amount	Decimal	Minimum conversion amount for bonus
bonusMaximumConversion Amount	Decimal	Maximum conversion amount for bonus
status	Integer	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
expiredDate	Long	Expiry date of bonus game (Unix time stamp in milliseconds)
updatedAt	Long	Date when player is registered or re-registered into bonus game
createdDate	Long	(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,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdDate": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdDate": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

2.2.4 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

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
bonus_id	Integer	Yes	Unique identity of bonus game
player_name	String	No	Unique identity of bonus game's unregistered player Note: Fuzzy matching is implemented
created_by	String	No	API or BackOffice user who registered player
status	Integer	No	Status of unregistered player's wallet: 0: Inactive (Cancelled) 1: Pending 2: Transferred

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_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: 0: Inactive (Cancelled) 1: Pending 2: Transferred
updatedAt	Long	Date when player is registered or re-registered into bonus game
createdAt	Long	(Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who registered player
updatedBy	String	

Example:

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


2.2.5 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

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
bonus_id	Integer	Yes	Unique identity of bonus game
player_name	String	No	Unique identity of bonus game's player Note: Fuzzy matching is implemented
from_time	Long	No	Time range of bonus game's start time
to_time	Long	No	(Unix time stamp in milliseconds)
from_created_time	Long	No	Time range of bonus game's created time
to_created_time	Long	No	(Unix time stamp in milliseconds)
status	Integer	No	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

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 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: G: Free Game B: Bonus
balanceAmount	Decimal	Player balance of unconverted bonus game, or converted amount of converted bonus game
bonusRatioAmount	Decimal	Wagering requirement for bonus game. Refer to the total amount that need to be completed
bonusMinimumConversion Amount	Decimal	Minimum conversion amount for bonus game
bonusMaximumConversion Amount	Decimal	Maximum conversion amount for bonus game
status	Integer	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
expiredDate	Long	Expiry date of bonus game (Unix time stamp in milliseconds)
updatedAt	Long	Date when player is registered or re-registered into bonus game
createdAt	Long	(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,
        "expiredDate": 1546991999000,
        "status": 2,
        "createdDate": 1546923615000,
        "updatedAt": 1546991999000
      },
      {
        "balanceId": 111252,
        "playerId": 7443,
        "playerName": "player2",
        "walletType": "G",
        "balanceAmount": 8.8,
        "bonusMinimumConversionAmount": 0,
        "bonusMaximumConversionAmount": 0,
        "expiredDate": 1546991999000,
        "status": 6,
        "createdDate": 1546918715000,
        "updatedAt": 1547028195000
      }
    ]
  },
  "error": null
}
```

2.2.6 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

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
player_name	String	Yes	Unique identity of bonus player
status	Integer Array	No	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&page_number=1&row_count=500&player_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	Integer	Unique identity of bonus game Note: Free game id is return for bonus 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. Refer to the total amount that need to be completed
bonusParentType	Integer	Type of bonus parent: B: Bonus game that was created normally G: Bonus game that was converted from free game
status	Integer	Status of bonus wallet: 0: Inactive (Cancelled) 1: Active 2: Expired 3: Converted 4: Completed 5: New 6: User Cancelled 7: Locked 8: Pending
createdDate	Long	Created date of bonus game (Unix time stamp in milliseconds)
expiredDate	Long	Expiry date of bonus game (Unix time stamp in milliseconds)
createdBy	String	API or BackOffice user who created or updated
updatedBy	String	bonus game

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
}
```

2.2.7 Create Bonus Game

To create new bonus game.

Request

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

HTTP method : POST

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 (Unix time stamp in milliseconds)
minimum_conversion_a mount	Decimal	No	Minimum conversion amount for bonus game NULL: no limit of conversion amount
maximum_conversion_a mount	Decimal	No	Maximum conversion amount for bonus game NULL: unlimited conversion amount
bonus_type	Integer	Yes	Bonus game wagering condition 0: Bonus 1: Cash 2: Bonus & Cash
bonus_ratio_amount**	Decimal	No	Wagering requirement for bonus game. Refer to the total amount that need to be completed
bonus_ratio**	Decimal	Yes	Wagering requirement for bonus game. Refer to the multiple of the turnover that need to be completed Turnover = bonus ratio x bonus amount
game_ids	Integer Array	Yes	Unique identity of games Note: Only games with same base bet are accepted
is_event	Boolean	Yes	Type of bonus game: True: Flexible False: Standard

player_name	String	No	Unique identity of player. System will create bonus game and assign to this player directly
transaction_id	String	Yes	Unique identity of transaction. Note: System will return correspond bonus ID for duplicate transaction

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

Example:

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

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
bonusId	Integer	Unique identity of bonus

Example:

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


2.2.8 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.

Request

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

HTTP method : POST

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:

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: 0: Failed 1: Success 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
}
```

2.2.9 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.

Request

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

HTTP method : POST

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 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 Value range: Value \geq 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 Value range: Value \geq 1 Default value: 0 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 Value range: Value \geq 1 Default value: 0 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:

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

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: 0: Failed 1: Success 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
}
```

2.2.10 Cancel Bonus Game

To cancel a created bonus game.

Request

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

HTTP method : POST

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&bonus_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.2.11 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

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player1&balance_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.2.12 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

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&pending_wallet_id=10919

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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


2.2.13 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

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

Example:

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

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
actionResult	Integer	Request results False: Failed True: Success

Example:

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

2.3 Tournament [Latest version: v2]

2.3.1 Get Tournaments List

To get list of tournaments.

Request

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

HTTP method : POST

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
status	Integer	No	Status of tournament: 0: Cancel 1: InProgress 2: New 3: End
game_ids	Integer Array	No	Unique identity of games
language	String	No	Response language for tournament information: en-US: English (Default) zh-CN: Chinese
start_time	Long	No	Time range of tournament's start time and end time (Unix time stamp in milliseconds)
end_time	Long	No	

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&status=1&game_ids=1&game_ids=2&language=zh-cn&start_time=1559145500000&end_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	String	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 (Unix time stamp in milliseconds)
subTournamentEndTime	Long	End time of sub-tournament (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 ranking
pointRatio	Decimal	Minimum point needed to be spent in tournament to enter ranking Note: For accumulation mode only
gameMode	Integer	Tournament mode: 1: Accumulation 2: Depletion 3: Hybrid (Accumulation + Depletion)
rankStatus	Integer	Tournament leader board status: 0: Disabled 1: Enabled 2: Cut off (Tournament ranking had been finalized)
status	Integer	Status of tournament: 0: Cancel 1: InProgress 2: New 3: End
startTime	Long	Start time of tournament (Unix time stamp in milliseconds)
endTime	Long	End time of tournament

		(Unix time stamp in milliseconds)
createdTime	Long	Created time or updated time of tournament
updatedTime	Long	(Unix time stamp in milliseconds)

Example:

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

2.3.2 Get Tournament Players List

To get list of tournament players.

Request

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

HTTP method : POST

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
player_names	String	No	Unique identity of tournament players
from_created_time	Long	No	Time range of when player was added into tournament. (Unix time stamp in milliseconds)
to_created_time	Long	No	

Example:

```
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	Long	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)
updatedAtTime	Long	Time of when tournament player's information was updated (Unix time stamp in milliseconds)
createdBy	Integer	API or BackOffice user who registered player into tournament

Example:

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


2.3.3 Get Tournament Top Rankings

To get tournament top rankings.

Request

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

HTTP method : POST

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
player_name	String	No	Unique identity of player

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&tournament_id=41&count=10&player_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	Integer	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)

Example:

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

2.3.4 Register Tournament Players

To add **multiple players** into a tournament.

Request

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

HTTP method : POST

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 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: True: Allow False: Disallow

Example:

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	Integer	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	Tournament status: 0: Cancel 1: InProgress 2: New 3: End

Example:

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

2.3.5 Get Tournament Cash Prize Reward List

To get tournament cash reward list.

Request

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

HTTP method : POST

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 time and end time
end_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 14 days
transaction_id	Long	No	Unique identity of reward transaction
tournament_id	Integer	No	Unique identity of tournament
player_name	String	No	Unique identity of tournament players
currency	String	No	Currency of tournament
page_number	Integer	No	Batch number of data return Default value: 1
row_count	Integer	No	Number of records for each batch Default value: 5000

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&start_time=1559145500000&end_time=1559145600000

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
totalCount	Integer	Total number of tournament records
totalPage	Integer	Total page 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 of 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 (Unix time stamp in milliseconds)
tournamentPlayerId	Integer	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

Example:

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

2.4 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}**

2.4.1 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 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.

Flow

- **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 required count (e.g.: 1500 records per request)
- If number of records returned are less than 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

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 Note: Value range 1500-5000
bet_type	Integer	Yes	Bet types of bet record: 1: Real game
row_version	Long	Yes	Updated time of data (Unix time stamp in milliseconds) Note: <ul style="list-style-type: none"> Set value to 1 for first call 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: 0: All (Default) 1: Non-last hand 2: Last hand 3: Adjusted

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&row_version=1346592723000&hands_status=0

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
parentBetId	Integer	Unique identity of parent bet
betId	Integer	Unique identity of child bet (Unique key value)
playerName	String	Unique identity of player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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
handsStatus	Integer	Status of hand: 1: Non-last hand 2: Last hand 3: Adjusted
rowVersion	Integer	Updated time of data (Unix time stamp in milliseconds)
betTime	Integer	Start time of current bet (Unix time stamp in milliseconds)
betEndTime	Integer	End time of current bet (Unix time stamp in milliseconds)

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

Example:

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

2.4.2 Get History for Specific Time

To get the bet history for a specific time range.

Notes

- Operator can retrieve the bet history for 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.

Flow

- **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 required count (e.g.: 1500 records per request)
- If number of records returned are less than 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

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 Note: Value range 1500-5000
bet_type	Integer	Yes	Bet types of bet record: 1: Real game
from_time	Long	Yes	Time range of the start time and end time of the bed records
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 40 days

Example:

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

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
parentBetId	Integer	Unique identity of parent bet
betId	Integer	Unique identity of child bet (Unique key value)
playerName	String	Unique identity of player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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
handsStatus	Integer	Status of hand: 1: Non-last hand 2: Last hand 3: Adjusted
rowVersion	Integer	Updated time of data (Unix time stamp in milliseconds)
betTime	Integer	Start time of current bet (Unix time stamp in milliseconds)
betEndTime	Integer	End time of current bet (Unix time stamp in milliseconds)

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

Example:

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


2.4.3 Get Single Player History

To get bet history of specific player.

Notes

- Operator can retrieve the bet history for 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

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: 1: Real game
start_time	Long	Yes	Time range of bet record's start time and end time
end_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 7 days

Example:

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
parentBetId	Integer	Unique identity of parent bet
betId	Integer	Unique identity of child bet (Unique key value)
playerName	String	Unique identity of player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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	Integer	Updated time of data (Unix time stamp in milliseconds)
betTime	Integer	Date of current record (Unix time stamp in milliseconds)
handsStatus	Integer	Status of hand: 1: Non-last hand 2: Last hand 3: Adjusted

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

Example:

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

2.4.4 Get Player's Daily Summary

To get player's daily bet history.

Notes

- Operator can retrieve the bet history for 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

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 Note: Value range 1500-5000
bet_type	Integer	Yes	Category of bet record: 1: Real game
row_version	Long	Yes	Updated time of data (Unix time stamp in milliseconds)
time_zone	Integer	No	Time zone: -5: UTC-5 -4: UTC-4 0: UTC (Default) 1: UTC+1 4: UTC+4 7: UTC+7 8: UTC+8

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&count=5000&bet_type=1&row_version=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: 1: Real player 2: Test player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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	Integer	Updated time of data (Unix time stamp in milliseconds)
summaryDate	Integer	Date of current record (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
}
```

2.4.5 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 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

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 Note: Value range 1500-5000
bet_type	Integer	Yes	Category of bet record: 1: Real game
row_version	Long	Yes	Updated time of data (Unix time stamp in milliseconds)
from_time	Long	Yes	Time range of bet record's start time and end time (Unix time stamp in milliseconds)
to_time	Long	Yes	Note: Value range 1 day -40 days
time_zone	Integer	No	Time zone: -5: UTC-5 -4: UTC-4 0: UTC (Default) 1: UTC+1 4: UTC+4 8: UTC+8

Example:

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: 1: Real player 2: Test player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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	Integer	Updated time of data (Unix time stamp in milliseconds)
summaryDate	Integer	Date of current record (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
}
```

2.4.6 Get Hands Summary

To get the game summary records of each hand.

Notes

- Operator can retrieve the bet history for 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

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 Note: Value range 1500-5000
bet_type	Integer	Yes	Bet types of bet record: 1: Real game
hands_status	Long	Yes	Status of hands: 0: All 1: Incomplete 2: Completed
row_version	Long	Yes	Updated time of data (Unix time stamp in milliseconds)

Example:

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	Integer	Unique identity of parent bet
playerName	String	Unique identity of player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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
handsStatus	Integer	Status of hand: 1: Non-last hand 2: Last hand 3: Adjusted
rowVersion	Integer	Updated time of data (Unix time stamp in milliseconds)
handStartTime	Integer	Start time of current hand (Unix time stamp in milliseconds)
handEndTime	Integer	End time of current hand (Unix time stamp in milliseconds)

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

Example:

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

2.4.7 Get Hourly Hands Summary

To get operator's hourly hands summary records.

Notes

- Operator can retrieve the bet history for 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/GetHandsSummaryHourly
HTTP method : POST

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 time and end time of bet records
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 40 days
currency	String	No	Currency used by player in the record
transaction_types	Integer Array	No	Category of transaction: 1: Cash 2: Bonus 3: Free game

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&from_time=1567296000000&to_time=1569801600000¤cy=cny&transaction_types=1&transaction_types=2

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
dateTime	Integer	Unique identity of parent bet
totalHands	Integer	Unique identity of player
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: 1: Cash 2: Bonus 3: Free game

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

Example:

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

```

        "transactionType": 3
      }
    ],
    "error": null
  }

```

2.4.8 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 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/GetPlayerBetsSummationTotal

HTTP method : POST

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: 1: Real game
from_time	Long	Yes	Time range of start time and end time of bet record
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 40 days
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: 0: Normal 1: Cash bonus
player_type	Integer	No	Type of player: 0: All 1: Real player 2: Test player
transaction_type	Integer	No	Type of transaction: 0: All 1: Cash

			2: Bonus 3: Free Game
time_zone	Integer	No	Time zone: -5: UTC-5 -4: UTC-4 0: UTC (Default) 1: UTC+1 4: UTC+4 8: UTC+8

Example:

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
currency	String	Currency used by player in the record
totalGames	Integer	Total count of game rounds
totalHands	Integer	Total count of game hands
totalBetAmountConverted	Decimal*	Converted total bet amount of player (currency as EUR)
totalWinLossAmountConverted	Decimal*	Converted total win loss amount of player (currency as EUR)
totalCompanyWinLossAmountConverted	Decimal*	Converted total win loss amount of company (currency as EUR)
totalJackpotRtpContributionAmountConverted	Decimal*	Converted total jackpot RTP contribution amount of records (currency as EUR)
totalJackpotContributionAmountConverted	Decimal*	Converted total jackpot contribution amount of records (currency as EUR)
totalJackpotWinAmountConverted	Decimal*	Converted total jackpot win amount of records (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": [
    {
      "currency": "CNY",
      "totalGames": 1232191,
      "totalHands": 1171198,
      "totalBetAmountConverted": 436315.92259,
      "totalWinLossAmountConverted": 476402.99206,
      "totalCompanyWinLossAmountConverted": -476402.99206,
      "totalJackpotRtpContributionAmountConverted": 0.01824,
      "totalJackpotContributionAmountConverted": 0.66614,
      "totalJackpotWinAmountConverted": 32000.13301,
      "totalBetAmount": 3417984.61,
      "totalWinLossAmount": 3732015.25799,
      "totalCompanyWinLossAmount": -3732015.25799,
      "totalJackpotRtpContributionAmount": 0.14268,
      "totalJackpotContributionAmount": 5.21856,
      "totalJackpotWinAmount": 250680.69
    },
    {
      "currency": "EUR",
      "totalGames": 75,
      "totalHands": 29,
      "totalBetAmountConverted": 266.0,
      "totalWinLossAmountConverted": 1439.8,
      "totalCompanyWinLossAmountConverted": -1439.8,
      "totalJackpotRtpContributionAmountConverted": 0.0,
      "totalJackpotContributionAmountConverted": 0.0,
      "totalJackpotWinAmountConverted": 0.0,
      "totalBetAmount": 266.0,
      "totalWinLossAmount": 1439.8,
      "totalCompanyWinLossAmount": -1439.8,
      "totalJackpotRtpContributionAmount": 0.0,
      "totalJackpotContributionAmount": 0.0,
      "totalJackpotWinAmount": 0.0
    }
  ],
  "error": null
}
```

2.4.9 Get Player's Unfinished Games

To get player unfinished hands

Notes

- Operator can retrieve the bet history for 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

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 tournament's start time and end time
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 7 days
game_ids	Integer Array	No	Unique identity of games
player_modes	Integer Array	No	Mode of game: 0: Normal 1: Cash bonus
player_type	Integer	No	Type of player: 0: All 1: Real player 2: Test player
transaction_type	Integer	No	Type of transaction: 0: All 1: Cash 2: Bonus 3: Free Game
time_zone	Integer	No	Time zone: -5: UTC-5 -4: UTC-4 0: UTC (Default) 1: UTC+1

			4: UTC+4 8: UTC+8
--	--	--	----------------------

Example:

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: 1: Cash 2: Bonus 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
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 (Unix time stamp in milliseconds)
handEndTime	Long	End time of the hand (Unix time stamp in milliseconds)

Example:

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

```
    "error": null  
}
```

2.4.10 Get Bet Details

To retrieve bet details for **multiple bets**

Notes

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

Request

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

HTTP method : POST

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	Integer	Yes	Unique identity of bets

Example:

operator_token=abcd&secret_key=abc123&bet_ids=1315486539764989952&bet_ids=1315486539764989960

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
parentBetId	Integer	Unique identity of parent bet
betId	Integer	Unique identity of child bet (Unique key value)
playerName	String	Unique identity of player
gameId	Integer	Unique identity of game
betType	Integer	Category of bet record: 1: Real game
transactionType	Integer	Category of transaction: 1: Cash 2: Bonus 3: Free game
platform	Integer	Platform of bet record (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
handsStatus	Integer	Status of hand: 1: Non-last hand 2: Last hand 3: Adjusted
rowVersion	Integer	Updated time of data (Unix time stamp in milliseconds)
betTime	Integer	Start time of current bet (Unix time stamp in milliseconds)
betEndTime	Integer	End time of current bet (Unix time stamp in milliseconds)

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

Example:

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

2.4.11 Get Failed Bets

To retrieve failed bets

Notes

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

Request

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

HTTP method : POST

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 time of enquiry
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 40 days
count	Integer	Yes	Number of records for each batch Note: Value range 1500-5000

Example:

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 (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
}
```

2.4.12 Pending Bets

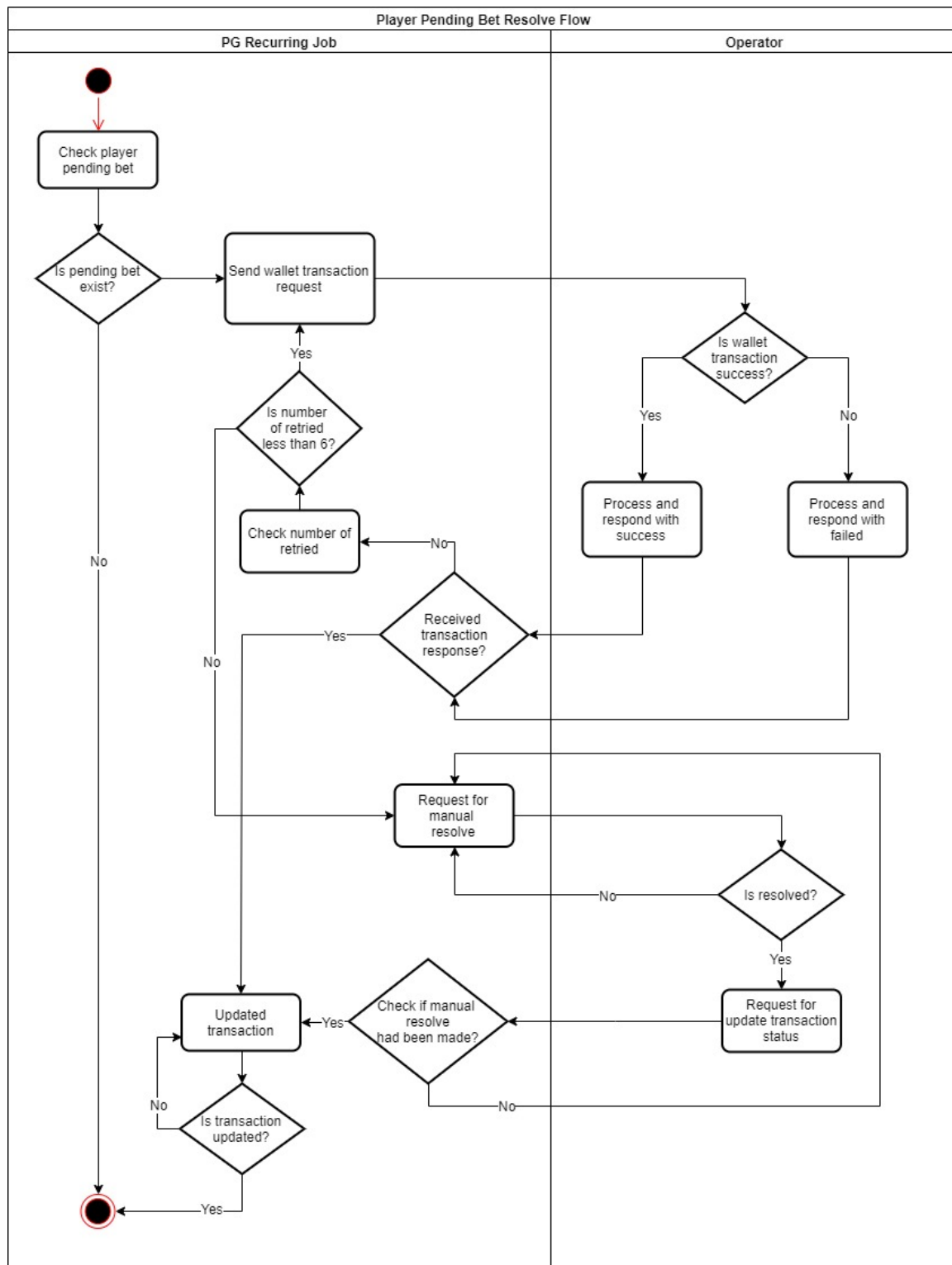
Sometimes pending bet may occur due to unforeseen circumstances during player game play for example, network latency, system failed, system timeout and so on.

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

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

Default interval of pending bet resolving time is set to 60 minutes. Pending bets which exceed 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 failed to resolve pending bets after several attempts. In such case, operator is required to resolve pending bet manually, and PG Soft will mark these bets status as complete after getting the confirmation from operator.



2.4.12.1 Get Pending Bets

To retrieve pending bets

Notes

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

Request

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

HTTP method : POST

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	Yes	Unique identity of player
from_time	Long	Yes	Time range of the start time and end time of enquiry
to_time	Long	Yes	(Unix time stamp in milliseconds) Note: Value range 1 day - 40 days
count	Integer	Yes	Number of records for each batch Note: Value range 1500-5000

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_names=testplayer1&player_names=testplayer2&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
gameId	String	Unique identity of game
betStatus	Integer	Status of bet 1: Pending

Example:

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

2.4.12.2 Manual Resolve Pending Bet

To resolve pending bet

Request

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

HTTP method : POST

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: True: Success bet False: Failed bet
game_id	Integer	Yes	Unique identity of game
balance_before_adjustment	Decimal	No	Player balance before adjustment (For success bet only)

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=testplayer1&bet_id=1325714346441543680&is_success_bet=true&game_id=1

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
resolveStatus	Boolean	Bet resolve result: True: Success False: Failed
betStatus	Integer	Status of bet 1: Pending 2: Failed 3: Success 4: Adjusted

Example:

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

2.4.13 Manual Reset Pending Game State

To reset pending game state

Request

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

HTTP method : POST

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	No	Operator session for player

Example:

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: True: Success False: Failed

Example:

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


2.5 Game [Latest version: v2]

2.5.1 Get Games List

To get list of games.

Request

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

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: en-us: English zh-cn: Chinese

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl¤cy=CNY&language=en-us

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: 0: Inactive 1: Active 2: Suspended Note: Game will only be accessible if game status and game release status are active
releaseStatus	Integer	Status of game for operator: 0: Inactive 1: Active 2: Suspended Note: Game will only be accessible if game status and game release status are active

gameLegalBetAmounts JSON Array

Parameter name	Data type	Description
gameId	Integer	Unique identity of game
gameTypeId	Integer	Type of game (Card game only): 0: None 1: Baccarat-Supersix 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,
      "gameLegalBetAmounts": [{
        "gameId": 1,
        "gameTypeId": 0,
        "legalBetAmounts": [
          {
            "coinSize": 0.03,
            "betMultiplier": 1,
            "baseBet": 30,
            "betAmount": 0.9
          },
          {
            "coinSize": 0.03,
            "betMultiplier": 2,
            "baseBet": 30,
            "betAmount": 1.8
          }
        ]
      }
    ],
    {
      "gameId": 2,
      "gameName": "Gem Saviour",
      "gameCode": "gem-saviour",
      "status": 1,
      "releaseStatus": 1,
      "gameLegalBetAmounts": [
        {
          "gameId": 1,
          "gameTypeId": 0,
          "legalBetAmounts": [{
            "coinSize": 0.03,
            "betMultiplier": 1,
            "baseBet": 30,
            "betAmount": 0.9
          },
          {
            "coinSize": 0.03,
            "betMultiplier": 2,
            "baseBet": 30,
            "betAmount": 1.8
          }
        ]
      }
    ]
  ],
  "error": null
}
```

2.5.2 Get Game Legal Bet Amounts

To get the legal bet amounts of games.

Request

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

HTTP method : POST

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:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl¤cy=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): 0: None 1: Baccarat-Supersix 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
}
```

2.5.3 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

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: 0: Inactive 1: Active
game_ids	Integer Array	Yes	Unique identity of games

Example:

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: true: Success false: Failed

Example:

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

2.6 Jackpot [Latest version: v1]

2.6.1 Get Jackpots List

To get list of jackpots.

Request

API URL : {PgSoftAPIDomain}/Jackpot/v1/Get
 HTTP method : POST

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 jackpot
game_id	Integer	No	Unique identity of game

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl¤cy=cny

Response

Body Parameters:

JSON Array

Parameter name	Data type	Description
jackpotName	String	Given name of jackpot
jackpotRoomId	String	Given room ID of jackpot
amount	Decimal*	Total amount of jackpot
jackpotType	Integer	Type of jackpot: 1: Grand 2: Mega 3: Major
poolId	Integer	Unique pool ID of jackpot pool
gameIds	Integer Array	Unique identity of games
lastUpdatedOn	Long	Updated date of jackpot (Unix time stamp in milliseconds)

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

Example:

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


}

2.7 Player [Latest version: v3]

2.7.1 Get Online Player Count

To get the total number of online players.

Request

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

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 Note: System will return list of online players for all games if parameter value is not provided

Example:

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 (Unix time stamp in milliseconds)
gameId	Integer	Unique identity for game
playerCount	Integer	Total number of online players
status	Integer	Status of player 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
}
```

2.7.2 Get Players Online Status

To get online status of last 10 players.

Request

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

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:

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 0: Offline 1: Online

Example:

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

2.7.3 Get Online Players

To get a list of online players.

Request

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

HTTP method : POST

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 Note: <ul style="list-style-type: none"> • Default value: 1 • Value range: 1-5000

Example:

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	Integer	Updated time of data (Unix time stamp in milliseconds)

Example:

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

2.7.4 Create Player

Create new player account in PG game system.

Note

Implementation of this API is optional as player account will create automatically for the first-time player access PG game

Request

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

HTTP method : POST

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 <ul style="list-style-type: none"> player name is not case sensitive Only alphabet, number, and "@", "-", " " symbols are allowed Note: Max 50 characters
nickname	String	No	Nickname of player Note: Max 50characters
currency	String	Yes	Currency of player

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123¤cy=EUR&nickname=player123

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
action_result	Integer	Status of the request 1: success 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's operation in progress

2.7.5 Kick Out Player

Kick out player from PG game system, yet player can re-login to games

Request

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

HTTP method : POST

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:

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 1: success 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's operation in progress

2.7.6 Suspend Player

Suspend player account, the suspended players cannot access PG games

Request

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

HTTP method : POST

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:

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 1: success 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's operation in progress

2.7.7 Resume Player

Resume suspended player account.

Request

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

HTTP method : POST

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:

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 1: success 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's operation in progress

2.7.8 Check Player Status

Check specific player's account status.

Note

This API is not to check online player status, just to check the status at PG. For online active player, please check at back office

Request

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

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:

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 0: Disabled 1: Active 2: Inactive/Deactivated 3: Suspended 4: Player's operation in progress

Example:

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

3. Appendix

3.1 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