

PG SOFT

Integration Document

Transfer Mode v2.2.1



Contents

Revision History	3
1. Overview	4
2. System Requirements	5
2.1 Game Dimensions	5
2.2 Hardware	5
2.3 Browser	5
3. Launch Game	6
3.1 Game authorization Modes	6
3.1.1 URL Scheme	7
3.1.2 Web Lobby	10
3.1.3 WebView	12
3.2 Game Launch Restrictions	15
4. Game Launch JS Library	16
4.1 Avoid Browser Pop-up Blockers	16
4.2 Implementation	17
5. Integration API	18
5.1 Operator Provided API	18
5.1.1 Verify Session	19
5.2 PG Provided API	21
5.2.1 Login Game	22
5.2.2 Check Balance	24
5.2.3 Transfer In and Transfer Out	26
5.2.4 Wallet Operation Restriction and Handling	31
5.2.5 Create Player (Optional)	32
5.2.6 Kick Out Player (Optional)	34
5.2.7 Suspend Player (Optional)	35
5.2.8 Resume Player (Optional)	36
5.2.9 Check Player Status (Optional)	37
5.2.10 Transfer All Out (Optional)	39
5.2.11 Get Single Transaction History (Optional)	41
5.2.12 Bet History Interpreter (Optional)	43
5.2.13 Others (Optional)	45
5.3 Integrate PG Game into Operator iOS Application	46
6. Resolve Pending Bet	47
7. Reset Game State	49
8. Restrictions	50
8.1 IP	50
8.2 Currency	50
9. Appendix	51
9.1 Error codes	51

9.2	Currencies	53
9.3	Bet Types	56
9.4	Transaction Types	57
9.5	Platforms.....	58
9.6	Games.....	59
9.7	Languages	61

Revision History

Version	Date	Description
2.2.0	2019-11-09	Revamp Version of document
2.2.1	2020-03-09	5.1.1 Verify Session <ul style="list-style-type: none"> Added note for operator_player_session parameter 5.2.10 Get Single Transaction History <ul style="list-style-type: none"> Amended API response
2.2.2	2020-09-24	3.2 Game Launch Restrictions <ul style="list-style-type: none"> Added new contents 5.2.4 Wallet Operation Restrictions and Handling <ul style="list-style-type: none"> Added new contents
2.2.3	2020-10-02	5.2.3 Transfer In and Transfer Out <ul style="list-style-type: none"> Added new error codes 5.2.3.1 Transfer In <ul style="list-style-type: none"> Added new error codes 5.2.3.2 Transfer Out <ul style="list-style-type: none"> Added new error codes 5.2.11 Get Single Transaction History <ul style="list-style-type: none"> Added new error codes

1. Overview

This document details the Integration of Transfer mode. So-called Transfer mode, means players need to make the transfer from their central wallet to PG games wallet before playing, and meanwhile, players need to make transfer from PG games wallet to their central wallet to withdraw their funds.

2. System Requirements

2.1 Game Dimensions

PG game are responsive H5 and displayed vertically, thus, minimum dimensions is required. Please refer to the following for game dimensions in iframe:

- Ideal dimensions : 1280 x 720
- Minimum dimensions : 540 x 960
- Maximum dimensions : 1080 x 2340

2.2 Hardware

	Android	iOS	Windows
Version of Operating System	5.0 or above	10.0 or above	7 or above
RAM	2GB	1GB	2GB
CPU	ARM Cortex-A9 or above	iPhone 6 or above	<ul style="list-style-type: none"> • 1 gigahertz (GHz) or faster • 32-bit (x86) or 64-bit (x64) processor • Dual core at least
Storage	2GB and above	2GB and above	20GB and above

2.3 Browser

- Microsoft Edge : version 15.15063 or above
- Firefox : version 44 or above
- Google Chrome : version 40 or above
- Safari : version 10 or above
- *others* : browser based on Chromium that equal to Google Chrome 40 or above

3. Launch Game

3.1 Game authorization Modes

PG supports three kinds of launch game authorization modes:

- **URL scheme** mode is for iOS, Android and browser, it uses the token parameters to authorize and login
- **Web lobby** mode is a responsive web page which uses token parameters to authorize and login.
- **WebView** mode is applicable to all platforms, it is the Integration mode which embeds Into the game Interface and uses username and password to authorize players to login.

3.1.1 URL Scheme

URL scheme is an ordinary method for game launch. Player is required to login via operator web site and launch game through operator game lobby page

Note

Operator is required to provide `verifySession`¹ API for operator player session authentication

Request

API URL : {PgSoftPublicDomain}/{GameCode}/index.html

HTTP method : GET

URL:

Parameter name	Data Type	Mandatory	Description
game_code	String	Yes	Unique code for each game ²

URL Parameters:

Parameter name	Data Type	Mandatory	Description
bet_type	Integer	Yes	Game launch mode ³
operator_token	String	Yes	Unique identity of operator
operator_player_session	String	Yes	Token generated by operator system Note: <ul style="list-style-type: none"> • Max 200 characters • Please encode the value with UrlEncode to avoid unexpected error
language	String	No	Display language of game ⁴ Default: en
time_elapsed	Integer	No	Operator time elapsed for game health reminder in seconds (Reality check ⁵)
reminder_interval	Integer	No	Reminder Interval for game health reminder in seconds (Reality check ⁶)
operator_param	String	No	Operator custom parameter, PG API will include parameter value upon verifying operator player session Note: Please encode the value with UrlEncode to avoid unexpected error
from	String	No	Game exit URL

¹ Please refer to [Verify Session](#) section for more information

² Please refer to [Games](#) section for complete list

³ Please refer to [Bet Types](#) section for complete list

⁴ Please refer to [Languages](#) section for complete list

⁵ Please refer to [Reality Check](#) section for more information

⁶ Please refer to [Reality Check](#) section for more information

			Default: Redirect to PG quit page Note: <ul style="list-style-type: none"> In Trial Game mode, value will be used for redirection of Real Game prompt Assign value to <i>PGGameCloseUrl</i> to close game window.
real_url	String	No	This URL will be used for redirection during Real Game prompt Default: Redirect to PG lobby Note: For trial game only
cached_t	String	No	To cache player session 0: Player session will be stored for PG game (Global) 1: Player session will be stored for each game respectively 2: Login to always call verifySession each time login. Player session will not be stored

Example:

https://m.pg-demo.com/diaochan/index.html?operator_token=abcd&operator_player_session=12345-abcd-1234-abcd-12345&bet_type=1

3.1.1.1 Reality Check

In URL scheme⁷, operator may include “time_elapsed” and “reminder_interval” parameter into launch game URL for reality check.

- time_elapsed : Time passed for first time reality check in second(s)
- reminder_interval : Reality check interval in second(s)

In use case

reminder_interval=300&time_elapsed=290

The first reality check reminder will show after 10 seconds (300s-290s=10s), and subsequence reminder will show for every 5 minutes

⁷ Please refer to [URL Scheme](#) section for more information

3.1.2 Web Lobby

PG web lobby is a responsive web page with all PG games which allow player to play PG games in one place. It provides brand new experience to player with beautiful, attractive, informative and effective UI design. We are recommended operator to Integrate PG web lobby into their web site.

Player is required to login via operator web site and launch PG game through PG web lobby.

Note

Operator is required to provide verifySession⁸ API for operator player session authentication

Request

API URL : {PgSoftPublicDomain}/web-lobby/{panel_type}
HTTP method : GET

URL:

Parameter name	Data Type	Mandatory	Description
panel_type	Integer	Yes	Display panel of web lobby at first load [NO VALUE]
			smartbot : Default panel (Smartbot)
			smartbot : Smartbot – In progress panel
			smartbot/history : Smartbot – Stopped panel
			tournament/open : Tournament – Open for entry panel
			tournament/process: Tournament – In progress panel
			tournament/end : Tournament – Ended panel
			games : Show only game

⁸ Please refer to [Verify Session](#) section for more information

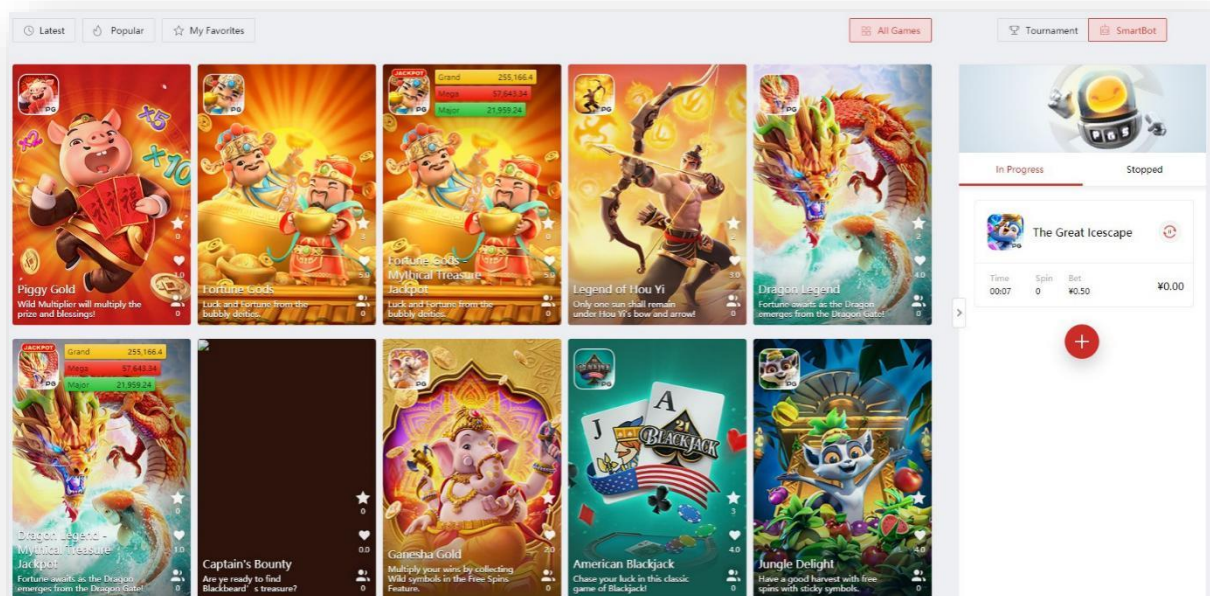
URL Parameters:

Parameter name	Data type	Mandatory	Description
operator_token	String	Yes	Unique identity of operator
operator_player_session	String	Yes	Token generated by operator system Note: <ul style="list-style-type: none"> • Max 200 characters • Please encode the value with <code>UrlEncode</code> to avoid unexpected error
language	String	No	Display language of web lobby ⁹ Default: Player device language
width	Integer	No	Page width of web lobby Default: Responsive Value range: Value ≥ 930

Example:

`https://public.pg-redirect.net/web-lobby/smartbot/?operator_token=abc123&operator_player_session=abc123-abc123&language=en`

Screenshot



⁹ Please refer to [Languages](#) section for complete list

3.1.3 WebView

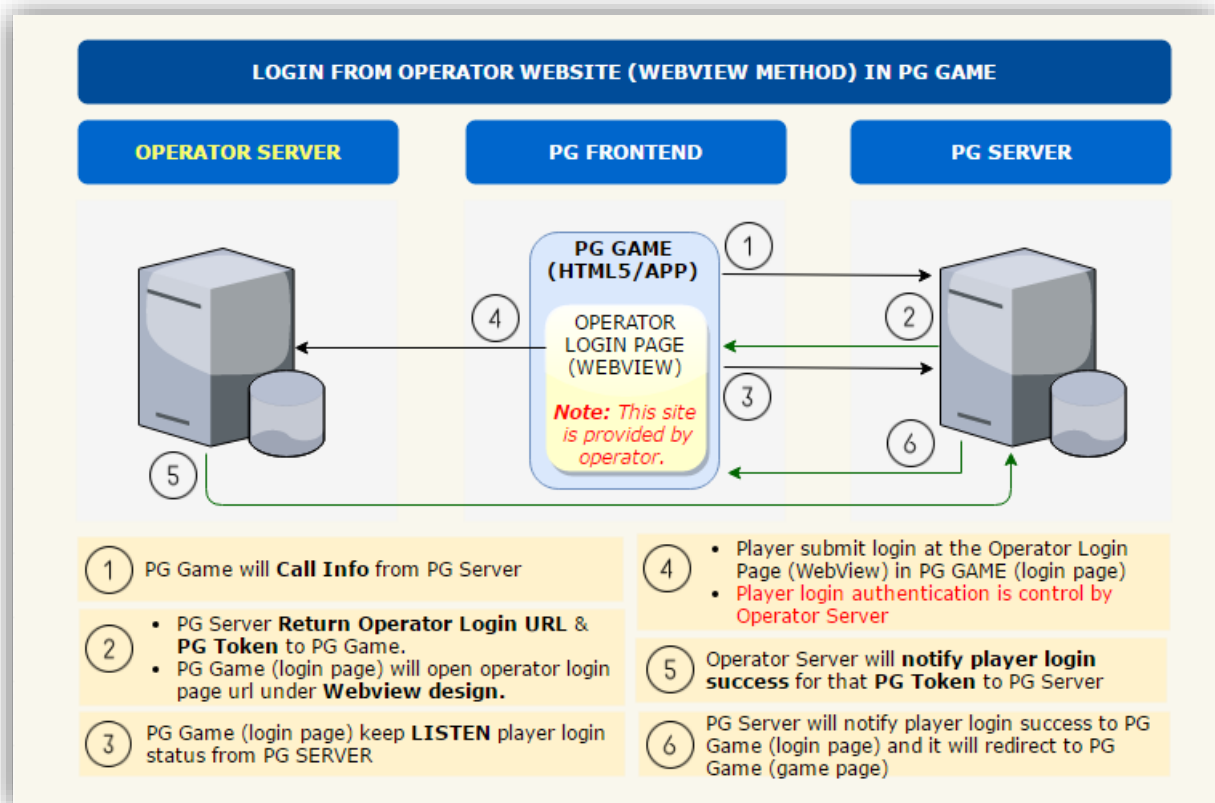
WebView login method provides seamless login experience to player. It is applicable to all platforms; player is required to log in and launch game through PG game client.

Note

- Operator is required to provide a login page (host in operator server) for player login
- Operator is required to call PG LoginGame API¹⁰ when player is login successfully

Login Flow

Operator is required to provide a login page, and this login page will be embedded into PG game client through WebView, to achieve the seamlessly login-effect. Player is required to login via operator login page and launch game through PG game client.



WebView login flow diagram

¹⁰ Please refer to [Login Game](#) section for more information

1. Operator Login Page

PG game client will open operator login page and concatenate with token parameter

Login Page URL : {OperatorLoginURL}

URL Parameters:

Parameter name	Data type	Mandatory	Description
pgsoft_token	String	Yes	Player token generated by PG system

Example:

`https://loginPage.operatordomain.com/login?pgsoft_token=pg_token_xxx;`

2. Notify Player Login Success

Player is required to login via operator login page.

If player login verification is successful, operator is required to call PG LoginGame API¹¹ to notify PG for the success login

¹¹ Please refer to [Login Game](#) section for more information

Screenshot

The screenshot shows a mobile application interface for a login page. At the top, there is a dark blue header bar with the text "Login" in orange on the left and "Reload" in orange on the right. Below the header, the page has a white background. On the left side, there is a "< Back" link in orange. In the center, the text "Log in" is displayed. Below this, the PG Pocket Games Software logo is shown. Under the logo, there are two input fields: the top one is labeled "Username" and the bottom one is labeled "Password". Below the password field, there is a link "Forgot Password?" in orange. At the bottom of the form area, there is a large orange button with the text "Log in" in white. Below the button, there is a line of text: "Don't have an account? Sign up", where "Sign up" is a link in orange.

Sample of WebView login page

3.2 Game Launch Restrictions

Multiple windows or multiple game launch is unsupported by PG games. Player will be locked and kick out of games, and encounter error when multiple game session is detected:

S1302XXXXXXXX: Unsuccessful Game Authorization

S1309XXXXXXXX: Player account been suspended

S1305XXXXXXXX: Server request failure

When encountering the above errors, player may close the game windows and relaunch game to unlock player account:

S1302XXXXXXXX: Please relaunch game

S1309XXXXXXXX: Player account will be unlocked automatically after 10 minutes, please relaunch game after 10 minutes

S1305XXXXXXXX: Please relaunch game

4. Game Launch JS Library

Operator is required to launch game by using PG JS library to provide the best gaming experience to player:

- Implementation of automatically adapting pop-up window according to the user's display size.
- PG games are all designed to be displayed vertically, the pop-up window format is consistent with product design.
- Achieving the optimal gaming Interface height will allow users to feel visually comfortable, in line with usage habits.
- Presenting the best in-game visual quality to the players, providing the best gaming experience

4.1 Avoid Browser Pop-up Blockers

The general rules that popup blockers will engage are:

- Invoked from JavaScript that is not invoked by direct user action
- Pop-up content are not from same domain as source

Operator is recommended to follow the instruction below to avoid browser pop-up blockers:

- Create a redirection page and redirect to game launch URL (URL scheme¹²)
- When player click the game launch button/link, use the PG JS function (PGSDK.launchGame) and launch the redirection page.

¹² Please refer to [URL Scheme](#) section for more information

4.2 Implementation

Step 1

Include the following line in the HTML <head> tags:

```
<script defer src="https://public.pg-redirect.net/sdk/main.min.js"></script>
```

Note:

Operator can self-host the JS library as well: <https://public.pg-redirect.net/sdk/main.min.js>

Step2

Operator is required to create self-host redirection page to bypass popup blocker on web browser

Sample code of redirect page:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script>
    window.location.replace("https://m.pg-redirect.net/diaochan/index.html?language=en&bet_type=1&operator_token=12345abcde&operator_player_session=abcde-12345-abcde-12345");
</script>
</head>
</html>
```

Step3

Use the following JS script to launch PG game. Operator is required to create self-host redirection page to bypass popup blocker on web browser

```
PGSDK.launchGame('{operator_redirection_URL}');
```

Parameters:

Parameter name	Data Type	Mandatory	Description
operator_redirection_URL	String	Yes	Operator self-host page which will redirect to game launch URL (URL scheme ¹³)

Example:

```
PGSDK.launchGame('https://[OperatorDomain]/redirect.html');
```

¹³ Please refer to [URL Scheme](#) section for more information

5. Integration API

5.1 Operator Provided API

Request

The API uses HTTP form methods and a RESTful endpoint structure. PG will send 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, operator is required to return an API response with HTTP status code 200.

API uses global JSON response format for all API methods, with the following content type:

Content-Type: `application/json`

Parameters:

Parameter name	Data Type	Description
<code>data</code> ¹⁴	JSON Object	Response information for API methods
<code>error</code>	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]"
  }
}
```

¹⁴ This field will show *null* value in error response

5.1.1 Verify Session

PG game system will call this operator API to check the player session token generated when player login the game.

Note

- This API is required for URL scheme or web lobby authorization mode¹⁵
- New player will be created automatically

Request

API URL : {OperatorAPIDomain}/VerifySession

HTTP method : POST

Body Parameters:

Parameter name	Data Type	Mandatory	Description
traceld	String	No	Identifier of the request
operator_token	String	Yes	Unique identity of operator
secret_key	String	Yes	Shared passphrase between PGSoft and operator
operator_player_session	String	Yes	Token generated by operator system <ul style="list-style-type: none"> • Max 200 characters • Please decode the value with <code>UrlDecode</code> to avoid unexpected error
ip	String	No	Player IP address
custom_parameter	String	No	<i>operator_param</i> values in URL schema ¹⁶
game_id	Integer	No	Unique identity of games

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&operator_player_session=a2b3c4d5e6f7g8&ip=1.1.1.1&custom_parameter=operatorparam123&game_id=1

¹⁵ Please refer to [URL Scheme](#) and Web Lobby section for login flow and details

¹⁶ Please refer to [URL Scheme](#) section for more information

Validation

Operator is required to perform validation to the following parameters:

Parameter name	Description	Expected behaviour
operator_token	Make sure the parameter value is exactly same as PG provided.	Return error when not match
secret_key		
operator_player_session	Make sure the operator_player_session is match with token generated in URL scheme ¹⁷	

Response

Body Parameters:

JSON Object

Parameter name	Data type	Mandatory	Description
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:

```
{
  "data": {
    "player_name": "player1",
    "nickname": "nickname",
    "currency": "USD"
  },
  "error": null
}
```

Error Response

Code	Message
1034	Invalid request
1200	Internal server error

¹⁷ Please refer to [URL Scheme](#) section for more information

5.2 PG Provided API

Request

The API uses HTTP form methods and a RESTful endpoint structure. Operator is required to send 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, PG 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
<code>data</code> ¹⁸	JSON Object	Response information for API methods
<code>error</code> ¹⁹	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]"
  }
}
```

¹⁸ This field will show *null* value in error response

¹⁹ This field will show *null* value in success API response. Please refer to [Error codes](#) section for error response format

5.2.1 Login Game

To notify PG server that the player's login has been successfully authorized.

Note

- This API is required for WebView authorization mode²⁰
- New player will be created automatically

Request

API URL : {PgSoftAPIDomain}/v1/Login/LoginGame

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_session	String	Yes	Token generated by PG
operator_player_session	String	Yes	Token generated by operator system Note: Max 200 characters
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
currency	String	Yes	Currency of player
nickname	String	No	Nickname of player Note: Max 50characters

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_session=EE9125CB-8EA2-42C5&operator_player_session=a2b3c4d5e6f7g8&player_name=player123¤cy=EUR&nickname=player123

²⁰ Please refer to [WebView](#) section for more information

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
player_name	String	Name of player
player_session	String	Session of player

Example:

```
{
  "data": {
    "player_name": "player1",
    "player_session": "28860417-4921-412A-AD2F-5988A41AA4BD"
  },
  "error": null
}
```

Error Response

Code	Message
1034	Invalid request
1200	Internal server error

5.2.2 Check Balance

Check specific player's balance

Request

API URL : {PgSoftAPIDomain}/v3/Cash/GetPlayerWallet

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
currencyCode	String	Currency of player
totalBalance	Decimal	Summation of all player wallet balance: <ul style="list-style-type: none"> • Cash balance • Bonus balance • Free game balance
cashBalance	Decimal	Player cash wallet balance Operator may refer this value for player balance
totalBonusBalance	Decimal	Player bonus wallet balance This will show all available bonus balance of player. It cannot be transfer out
freeGameBalance	Decimal	Player free game wallet balance This will show all available free game balance of player. It cannot be transfer out
bonuses	JSON Array	Player bonus details
freeGames	JSON Array	Player free game details

Example:

```
{
  "data": {
    "currencyCode": "CNY",
    "totalBalance": 997.00,
    "cashBalance": 997.00,
    "totalBonusBalance": 0.0,
    "freeGameBalance": 0.0,
    "bonuses": [],
    "freeGames": []
  },
  "error": null
}
```

Error Response

Code	Message
3001	Value cannot be null
3005	Player wallet doesn't exist

5.2.3 Transfer In and Transfer Out

PG system are implementing **idempotent operations** for all API operation. Operator is required to **resend** the request by using same transfer_reference for the following scenarios:

- Received incorrect response (incorrect response format) from PG system
- Received response with error code other than 3001, 3005 or 3100
- Did not receive response from PG system

5.2.3.1 Transfer In

Transfer into specific player's cash wallet

Note

- Transfer_reference is the unique identifier for every transaction. Operator is required to send new transfer_reference for every transfer request.
- For duplicate request, PG will not update player wallet balance, and return success response for duplicate request.

Request

API URL : {PgSoftAPIDomain}/v3/Cash/TransferIn

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
amount	Decimal	Yes	Transfer in amount (2 decimal place) Note: Min 0.01
transfer_reference	String	Yes	Transfer reference, generated by operator to make sure the transfer is unique Note: Max 50 characters
currency	String	Yes	Currency of player

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&amount=0.30&transfer_reference=ref1530879795000¤cy=EUR

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
transactionId	String	Unique transaction identifier
balanceAmountBefore	Decimal	Player balance before transaction
balanceAmount	Decimal	Player balance after transaction
amount	Decimal	Transaction amount

Example:

```
{
  "data": {
    "transactionId": 35509540,
    "balanceAmountBefore": 1000.00,
    "balanceAmount": 1100.00,
    "amount": 100.00
  },
  "error": null
}
```

Error Response

Code	Message
3001	Value cannot be null
3005	Player wallet does not exist
3100	Wallet transaction failed
3101	Wallet transaction in progress, please retry

5.2.3.2 Transfer Out

Transfer out from specific player's cash wallet

Note

- Transfer_reference is the unique identifier for every transaction. Operator is required to send new transfer_reference for every transfer request.
- For duplicate request, PG will not update player wallet balance, and return success response for duplicate request.

Request

API URL : {PgSoftAPIDomain}/v3/Cash/TransferOut

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
amount	Decimal	Yes	Transfer in amount (2 decimal place) Note: Min 0.01
transfer_reference	String	Yes	Transfer reference, generated by operator to make sure the transfer is unique Note: Max 50 characters
currency	String	Yes	Currency of player

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&amount=0.30&transfer_reference=ref1530879795000¤cy=EUR

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
transactionId	String	Unique transaction identifier
balanceAmountBefore	Decimal	Player balance before transaction
balanceAmount	Decimal	Player balance after transaction
amount	Decimal	Transaction amount

Example:

```
{
  "data": {
    "transactionId": 35509540,
    "balanceAmountBefore": 1000.00,
    "balanceAmount": 1100.00,
    "amount": 100.00
  },
  "error": null
}
```

Error Response

Code	Message
3001	Value cannot be null
3005	Player wallet does not exist
3100	Wallet transaction failed
3101	Wallet transaction in progress, please retry

5.2.4 Wallet Operation Restriction and Handling

Performing wallet transaction and launching game at the same time is prohibited to avoid missing player balance. In such case, PG will lock player and return error response for wallet API request:

```
1305: Invalid player - Player is locked: [playername]
```

Error Handling

1305

Operator shall resend wallet transaction request while receiving this error.

PG system are implementing **idempotent operations** for all API operation. Operator is required to **resend** the request by using same transfer_reference to avoid multiple balance transfer.

3004

PG will return this error when there is unhandled error happened when creating player. Operator may recreate player wallet by External API or relaunching PG game to create player automatically.

5.2.5 Create Player (Optional)

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}/v1/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)

5.2.6 Kick Out Player (Optional)

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

Request

API URL : {PgSoftAPIDomain}/v1/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

5.2.7 Suspend Player (Optional)

Suspend player account, the suspended players can't access PG games

Request

API URL : {PgSoftAPIDomain}/v1/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

5.2.8 Resume Player (Optional)

Resume suspended player account.

Request

API URL : {PgSoftAPIDomain}/v1/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

5.2.9 Check Player Status (Optional)

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}/v1/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: Inactive 1: Active 3: Suspended

Example:

```
{
  "data": {
    "player_name": "player123",
    "status": 0
  },
  "error": null
}
```

Error Response

Code	Message
1034	Invalid request
1035	Operation failed
1200	Internal server error
1204	Invalid operator

5.2.10 Transfer All Out (Optional)

Transfer out all wallet amount from specific player's cash wallet

Note

- Transfer_reference is the unique identifier for every transaction. Operator is required to send new transfer_reference for every transfer request.
- For duplicate request, PG will not update player wallet balance, and return success response for duplicate request.

Request

API URL : {PgSoftAPIDomain}/v3/Cash/TransferAllOut

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
amount	Decimal	Yes	Transfer in amount (2 decimal place) Note: Min 0.01
transfer_reference	String	Yes	Transfer reference, generated by operator to make sure the transfer is unique Note: Max 50 characters
currency	String	Yes	Currency of player

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&amount=0.30&transfer_reference=ref1530879795000¤cy=EUR

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
transactionId	String	Unique transaction identifier
balanceAmountBefore	Decimal	Player balance before transaction
balanceAmount	Decimal	Player balance after transaction
amount	Decimal	Transaction amount

Example:

```
{
  "data": {
    "transactionId": 35509540,
    "balanceAmountBefore": 1000.00,
    "balanceAmount": 1100.00,
    "amount": 100.00
  },
  "error": null
}
```

Error Response

Code	Message
3001	Value cannot be null
3005	Player wallet doesn't exist

5.2.11 Get Single Transaction History (Optional)

Get single transaction history of specific player searching by the transfer reference, player username.

Request

API URL : {PgSoftAPIDomain}/v3/Cash/GetSingleTransaction
 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
transfer_reference	String	Yes	Transfer reference, generated by operator to make sure the transfer is unique Note: Max 50 characters

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijkl&player_name=player123&transfer_reference=ref1530879795000

Response

Body Parameters:

JSON Object

Parameter name	Data type	Description
transactionId	String	Unique transaction identifier
playerName	String	Unique identity of players
currencyCode	String	Currency of player
transactionType	Integer	Transaction type: 100: TransferInCash 200: TransferOutCash
transactionAmount	Decimal	Transaction amount
transactionFrom	Decimal	Wallet balance before transaction
transactionTo	Decimal	Wallet balance after transaction
transactionDateTime	Long	Transaction datetime (Unix time stamp in milliseconds)

Example:

```
{
  "data": {
    "transactionId": 35595943,
    "playerName": "player123",
    "currencyCode": "EUR",
    "transactionType": 100,
    "transactionAmount": 1000000.00000,
    "transactionFrom": 0.00,
    "transactionTo": 1000000.00,
    "transactionDateTime": 1522984865000
  },
  "error": null
}
```

Error Response

Code	Message
3040	Transaction does not exist

5.2.12 Bet History Interpreter (Optional)

To **show** player bet result and details for specific bet

Step 1

Before getting betting detail, operator is required to call this API to get operator's session (Session timeout: 30 minutes).

Request

API URL : {PgSoftAPIDomain}/v1/Login/LoginProxy

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

Example:

operator_token=abc123&secret_key=a1b25cde5f3gh46ijk1

Response

Body Parameters:

JSON Object

Parameter name	Data Type	Description
operator_session	String	Operator session

Example:

```
{
  "data": {
    "operator_session": E4ECB5CF-0BC6-4DA8-8551-8526F48AB9CD
  },
  "error": null
}
```

Step 2

Get betting detail by

API URL : {PgSoftPublicDomain}/redirect.html

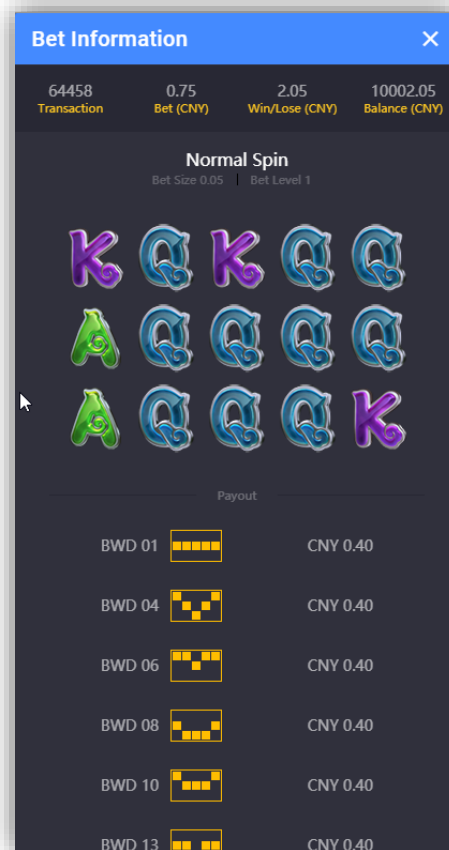
HTTP method : Get

Body Parameters:

Parameter name	Data Type	Description
t	String	Operator session from Step 1
psid	String	Parent Bet ID
sid	String	Bet ID
gid	String	Game ID
lang	String	Language: <ul style="list-style-type: none"> en (default) zh
type	String	Fixed value: operator

Example:

<https://public.pg-redirect.net/history/redirect.html?t=E4ECB5CF-0BC6-4DA8-8551-8526F48AB9CD&psid=12345&sid=12345&gid=1&lang=en&type=operator>



Screenshot of betting detail

5.2.13 Others (Optional)

PG provides several APIs for different modules and it is optional for integration. Operator may choose to integrate based on requirements:

- **Bet History** : To get player bet history
- **Game List** : To get latest game list and available bet size
- **Free Game** : To perform several actions (For example create free game, assign player into free game, cancel free game, etc.) through API
- **Bonus** : To perform several actions (For example create bonus, assign player into bonus, cancel bonus, etc.) through API.
- **Jackpot** : To get jackpot information through API
- **Tournament** : To get tournament list, add player into tournament and get tournament ranking through API

5.3 Integrate PG Game into Operator iOS Application

Operator's application has to meet the following requirements in order to launch PG game in application client

- iOS 8.0 or above
- WKWebView

PG game will prompt a "Poor Visit Experience" message if it does not meet the minimum requirements. Player is allowing to continue launching game, but it may not guarantee a stable, and high-performance experience of the PG games.

To ensure application client pass the PG game requirement checking, operator is required to include the following codes when initializing WKWebView component in application:

```
WKUserContentController *controller = [[WKUserContentController alloc] init];
[controller addScriptMessageHandler: self name: @"Could be any string value"];

WKWebViewConfiguration *configuration = [[WKWebViewConfiguration alloc] init];
configuration.userContentController = controller;

WKWebView *webView = [[WKWebView alloc] initWithFrame: CGRectZero
configuration:
configuration];
webView.uiDelegate = self;
self.view = webView;
```

This is to construct a WKUserContentController object and set its userContentController property on the configuration. Please take note that the userContentController must have at least one scriptMessageHandler (assign through addScriptMessageHandler)

For more details regarding WKWebView, please refer to [Apple Developer Documentation](#)

6. Resolve Pending Bet

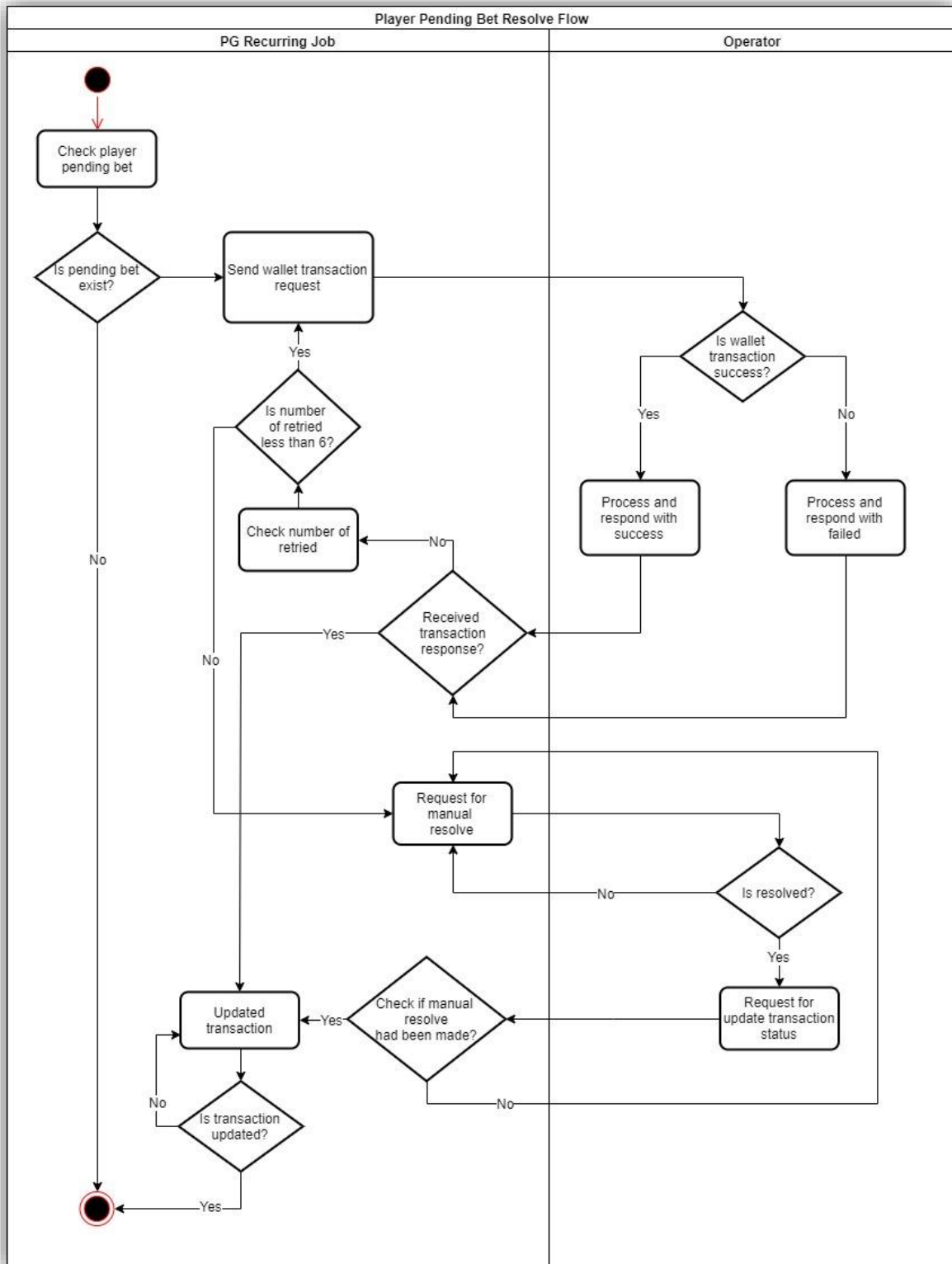
All Sometimes pending bet may occur due to unforeseen circumstances during player game play for example, network latency, system failed, system timeout and so on.

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.



Player Pending Bet Resolve Flow

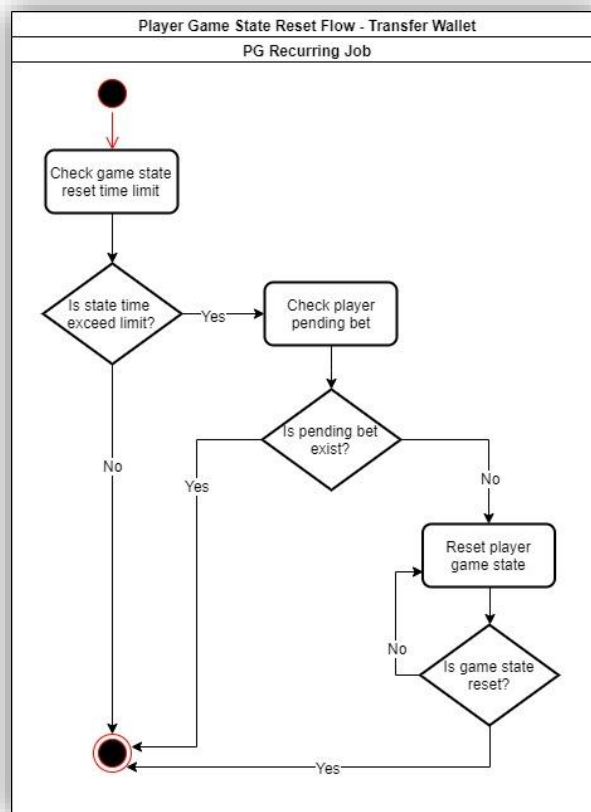
7. Reset Game State

All games support resume feature which allow returning players to continue a game at their last game state from any device. For example, player will be able to continue game with bonus state few days after even player had closed game the time when player won bonus state.

However, player game state will be reset for every 90 days by default. This interval is customizable, and every operator is allowed to have their own game state reset interval time. System will clear all finished and unfinished game progress, and player will not be able to continue a game after it is reset.

For transfer wallet mode, every player game state will be reset for every 90 days by default:

- PG Soft system will reset player game state automatically if there is no pending bet for player



Player Game State Reset Flow for Transfer Wallet

8. Restrictions

8.1 IP

Players from the following restriction area are not able to access PG game:

- Latvia
- Malaysia
- Singapore
- Taiwan
- United States of America

8.2 Currency

Please find the following for the currencies not supported by PGSoft:

- HKD
- MOP
- MYR
- SGD
- TWD
- PHP

9. Appendix

9.1 Error codes

Error code	Description
1034	Invalid request
1035	Operation Failed
1200	Internal server error
1204	Invalid operator
1300	Invalid player session
1301	Player session token is empty
1302	Invalid player session
1303	Server error occurs
1305	Invalid player
1306	Player is blocked to access current game
1307	Invalid player session
1308	Player session is expired
1309	Player is inactive
1310	Failed to verify operator player session
1400	Game is under maintenance
1401	Game is inactive
1402	Game does not exist
3001	Value cannot be null
3004	Player doesn't exist
3005	Player wallet doesn't exist
3006	Player wallet already exists
3009	Free game doesn't exist
3013	Out of the balance amount to transfer out
3014	Free game cannot be cancelled
3019	Not enough free game
3021	No bet exists
3022	Bet already pay-out
3030	Free game expired
3031	Free game already converted
3032	Bet already existed
3033	Bet failed
3034	Pay-out failed
3035	Invalid multiplier
3036	Not enough balance to convert
3040	Transaction does not exist

3202

Not enough cash balance to bet

9.2 Currencies

Currency Code	Currency Name	Base Unit
AED	Dirham des Émirats arabes unis	1
ALL	Albanian Lek	1
AMD	Armenian Dram	1
ARS	Argentine Peso	1
AUD	Australian Dollar	1
AZN	Azerbaijani Manat	1
BAM	Bosnian Convertible Marka	1
BDT	Taka bangladais	1
BGN	Lev bulgare	1
BIF	Burundian Franc	1000
BND	Brunei Dollar	1
BOB	Boliviano bolivien	1
BRL	Brazil Real	1
CAD	Canadian Dollar	1
CHF	Swiss Franc	1
CLP	Chilean Peso	1
CNY	Chinese Yuan	1
COP	Colombian Peso	1000
CRC	Colon costaricain	1
CSD	Serbian Dinar	1
CZK	Czech Koruna	1
DKK	Danish Krone	1
DOP	Peso dominicain	1
DZD	Dinar algérien	1
EGP	Livre égyptienne	1
EUR	Euro	1
GBP	British Pound	1
GEL	Georgian Lari	1
GHS	Ghanaian Cedi	1
GTQ	Quetzal guatémaltèque	1
HNL	Honduran Lempira	1
HRK	Croatian Kuna	1
HUF	Hungarian Forint	1
IDR	Indonesian Rupiah	1000
ILS	Shekel israélien	1
INR	Indian Rupee	1
IQD	Dinar irakien	1000

IRR	Iranian Rial	1000
ISK	Couronne islandaise	1
JPY	Japanese Yen	1
KES	Kenyan Shilling	1
KGS	Som du Kirghizistan	1
KHR	Cambodian Riel	1000
KRW	South Korean Won	1000
KZT	Kazakhstani Tenge	1
LBP	Livre libanaise	1000
LKR	Roupie sri lankaise	1
LYD	Dinar libyen	1
MAD	Moroccan Dirham	1
MBTC	Milli Bitcoin	1
MKD	Dinar macédonien	1
MMK	Burmese Kyat	1000
MNT	Mongolian Tughrik	1000
MWK	Malawian Kwacha	1
MXN	Mexican Peso	1
MZN	Mozambican Metical	1
NGN	Nigerian Naira	1
NIO	Cordoba nicaraguayen	1
NOK	Norwegian Krone	1
NZD	New Zealand dollar	1
PAB	Balboa panaméen	1
PEN	Peruvian Sol	1
PLN	Poland Złoty	1
PYG	Guarani paraguayen	1000
QAR	Rial qatari	1
RON	Romanian Leu	1
RSD	Serbian Dinar	1
RUB	Russian Ruble	1
SAR	Rial saoudien	1
SCR	Seychellois Rupee	1
SDG	Livre soudanaise	1
SEK	Swedish Krona	1
SVC	Colon salvadorien	1
SYR	Livre syrienne	1
THB	Thai Baht	1
TND	Dinar tunisien	1
TRY	Turkish Lira	1
TZS	Tanzanian Shilling	1000

UAH	Ukrainian Hryvnia	1
UBTC	United Bitcoin	1
UGX	Uganda Shilling	1000
USD	United States Dollar	1
UYU	Peso uruguayen	1
VND	Vietnamese Dong	1000
XAF	Central African CFA Franc BEAC	1
XOF	CFA Franc	1
YER	Rial yéménite	1
ZAR	Rand sud-africain	1
ZMW	Zambian Kwacha	1

9.3 Bet Types

Bet Type	Game Mode
1	Real game
2	Trial game
3	Tournament game

9.4 Transaction Types

Code	Transaction type
1	Cash
2	Bonus
3	Free game

9.5 Platforms

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

9.6 Games

Game Type	Game	Game Id	Game Code
Slot Game	Game Lobby	0	lobby
	Honey Trap of Diao Chan	1	diaochan
	Gem Saviour	2	gem-saviour
	Fortune Gods	3	fortune-gods
	Summon & Conquer	4	summon-conquer
	Medusa 2	6	medusa2
	Medusa 1	7	medusa
	Peas Fairy	8	peas-fairy
	Wizdom Wonders	17	wizdom-wonders
	Hood vs Wolf	18	hood-wolf
	Steampunk	19	steam-punk
	Tiki Go	21	tiki-go
	Win Win Won	24	win-win-won
	Plushie Frenzy	25	plushie-frenzy
	Tree of Fortune	26	fortune-tree
	Restaurant Craze	27	restaurant-craze
	Hotpot	28	hotpot
	Dragon Legend	29	dragon-legend
	Hip Hop Panda	33	hip-hop-panda
	Legend of Hou Yi	34	legend-of-hou-yi
	Mr. Hallow-Win	35	mr-hallow-win
	Prosperity Lion	36	prosperity-lion
	Santa's Gift Rush	37	santas-gift-rush
	Gem Saviour Sword	38	gem-saviour-sword
	Piggy Gold	39	piggy-gold

	Jungle Delight	40	jungle-delight
	Symbols of Egypt	41	symbols-of-egypt
	Ganesha Gold	42	ganesha-gold
	Three Monkeys	43	three-monkeys
	Emperor's Favour	44	emperors-favour
	Tomb of Treasure	45	tomb-of-treasure
	Double Fortune	48	double-fortune
	Journey to the Wealth	50	journey-to-the-wealth
	Wild Inferno	52	wild-inferno
	The Great Icescape	53	the-great-icescape
	Captain's Bounty	54	captains-bounty
	Ninja vs Samurai	59	ninja-vs-samurai
	Leprechaun Riches	60	leprechaun-riches
	Flirting Scholar	61	flirting-scholar
	Muay Thai Champion	64	muay-thai-champion
Card Game	Joker Wild	10	joker-wild
	American Blackjack	11	blackjack-us
	European Blackjack	12	blackjack-eu
	Baccarat Deluxe	31	baccarat-deluxe

9.7 Languages

Code	Language
en	English (default)
da	Danish
de	German
es	Spanish
fi	Finnish
fr	French
id	Indonesian
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
no	Norwegian
pl	Polish
pt	Portuguese
ro	Romanian
ru	Russian
sv	Swedish
th	Thai
tr	Turkish
vi	Vietnamese
zh	Chinese