

BETCONSTRUCT

Casino Third Party Games Integration to RGS API 1.1

**Quick Reference
Doc rev 1.05**

Revision History

Rev #	Revision Date	Description	Approved by	Corrected by
1.01	12/10/2017	Inputs proofreading. Document creating.	E. Mkrtchyan	Igor Mouhsian
1.02	17/04/2019	Ch. 6 modified, § 6.1 added, Ch. 7 FSDeposit added, Ch. 8 Front End added, Ch.9 Free Spins added	E. Mkrtchyan	A. Korkotyan
1.03	25/04/2019	§ 9.1 parameter descriptions edited	A. Kunjryan	A. Korkotyan
1.04	05/06/2019	§ 9.1: a new parameter 'operatorCode' added	A. Harutyunyan	I. Mukhsian
1.05	02/08/2019	Contents was shifted. Ch 8 JackpotDeposit and §8.1 Error Codes inserted.	A. Harutyunyan	I. Mukhsian

Table of Contents

Revision History.....	2
Introduction	4
1 GetPlayerInfo.....	4
1.1 GetPlayerInfo request example.....	5
2 Withdraw.....	6
2.1 Error codes.....	6
3 Deposit.....	7
3.1 Error codes.....	7
4 Withdraw/Deposit.....	8
4.1 Error codes.....	8
5 Rollback Transaction.....	9
5.1 Error codes.....	10
6 FSWithdraw	11
6.1 Error Codes	11
7 FSDeposit	12
7.1 Error Codes	12
8 JackpotDeposit	13
8.1 Error Codes	13
9 Front End	14
10 Free Spins.....	14
10.1 Free Spin Creating	15

Introduction

This API documentation contains a description of parameters used for third party games integration into Remote Gaming Server (RGS).

The HTTPS POST method is used for all requests. The Content-Type of request is application/json. The following parameters are general for all requests

- **time** – request time, datetime string, with "DD-MM-YYYY HH:MM:SS" format
- **data** – JSON-encoded data, requested API method parameters
- **hash** – MD5 checks the sum of concatenated string, which consists of **PrivateKey** (given to partner) **time** and **data**.

1 GetPlayerInfo

Required parameter of data:

- **token** – identifies the player

Response parameters:

- **result** – request result (true if successful and false if failed)
- **err_desc** – error description (if "result" is false)
- **err_code** – error code (if "result" is false)
- **currencyId** – ISO currency code
- **totalBalance** – player's balance by specified currency
- **nickName** – player's nickname
- **gender** – 1 - male, 0 - female
- **country** – player's country
- **userID** – player's id is unique

1.1 GetPlayerInfo request example

```
{"time":"20-01-2016 13:1:15", "data": {"token": "d201dee2-3ce7-45d5-a0d3-e3ce6b6b357c"}, "hash":"ebf1a262ba693690d7be03755718aa5d"}
```

Error codes

err_code	description
8	Authentication Failed
34	Wrong Currency
84	Game is Blocked
106	Token Expired
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

2 Withdraw

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- gameId – identifies game
- currencyId – ISO currency code
- betAmount – amount to debit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

2.1 Error codes

err_code	description
4	Wrong Bet Amount
7	Wrong Game ID
8	Authentication Failed
21	Not Enough Money
29	Player Is Blocked
34	Wrong Currency
84	Game is Blocked
104	Transaction is already exist
106	Token Expired
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

3 Deposit

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- gameId – identifies game
- currencyId – ISO currency code
- betAmount – amount to debit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

3.1 Error codes

err_code	description
7	Wrong Game ID
8	Authentication Failed
34	Wrong Currency
63	Wrong Win Amount
84	Game is Blocked
104	The transaction is already existing
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

4 Withdraw/Deposit

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- gameId – identifies game
- currencyId – ISO currency code
- betAmount – amount to debit
- winAmount – amount to the credit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

4.1 Error codes

err_code	description
4	Wrong Bet Amount
7	Wrong Game ID
8	Authentication Failed
21	Not Enough Money
29	Player Is Blocked
34	Wrong Currency
63	Wrong Win Amount
84	Game is Blocked
104	Transaction is already existing
106	Token Expired
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

5 Rollback Transaction

Required parameters of data:

- token – identifies a player
- transactionId – the id of the transaction to rollback
- gameId – identifies game

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

5.1 Error codes

err_code	description
8	Authentication Failed
21	Not Enough Money
34	Wrong Currency
104	The transaction already exists
105	The transaction is already canceled
107	Transaction not found
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

5.1 Error codes

Request Parameters

Parameter	Type
time	string
hash	string
token	string
transactionId	string
roundId	string
gameId	string
currencyId	string
betAmount	decimal
winAmount	decimal
betInfo	string

Response Parameters

Parameter	Type
currencyId	string
totalBalance	decimal
nickName	string
gender	int
country	string
userId	int
transactionId	long
balance	decimal

6 FSWithdraw

The request should be implemented for Free Spin if the withdraw is mandatory on Provider side during Free Spins.

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- gameId – identifies game
- currencyId – ISO currency code
- betAmount – amount to debit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

6.1 Error codes

err_code	description
4	Wrong Bet Amount
7	Wrong Game ID
8	Authentication Failed
21	Not Enough Money
29	Player is Blocked
34	Wrong Currency
84	Game is Blocked
104	Transaction already exists
106	Token Expired
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

7 FSDeposit

The request is mandatory for Free Spin. The total win of Free Spins should be sent by this request.

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- gameId – identifies game
- currencyId – ISO currency code
- betAmount – amount to debit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

7.1 Error codes

err_code	description
7	Wrong Game ID
8	Authentication Failed
34	Wrong Currency
63	Wrong Win Amount
84	Game is Blocked
104	Transaction already exists
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

8 JackpotDeposit

The request is mandatory for Jackpot wins. By this request should be sent the Jackpot wins.

Required parameters of data:

- token – identifies a player
- transactionId – unique key, in order to identify a single financial transaction
- roundId – identifies game cycle (can be null)
- currencyId – ISO currency code
- betAmount – amount to debit
- betInfo – (can be used if the additional data should be sent)

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- transactionId – unique key which identifies the single financial transaction (RGS transaction ID)
- balance – player balance after transaction

8.1 Error codes

err_code	description
8	Authentication Failed
34	Wrong Currency
63	Wrong Win Amount
84	Game is Blocked
104	Transaction is already exist
114	Token not Found
130	General Error
200	Incorrect Parameters Passed

9 Front End

Game Launch URLs for demo and real modes.

Demo mode

```
https://YOUR-DOMAIN.COM/xxxxxxxxx?  
mode=demo&gameID=YOUR_GAME_ID&language=eng
```

Real mode

```
https://YOUR-DOMAIN.COM/xxxxxxxxx?  
mode=real_play&gameID=YOUR_GAME_ID&token= PLAYER_TOKEN&language=eng
```

10 Free Spins

Free Spins Bonus is a promotional tool, which allows the player to play a specified number of free rounds in a game. Free Spins are always played with a minimal bet specified for each game and currency and maximal lines. During Free Spins gameplay, bets are not deducted from the player's balance, but all wins, based on the wagering factor, are collected for the Free Spins in the Provider's system.

After all free spins are played, including wagering, if any, the Provider should send the request for adding money to the player's balance.

The implementation of the Free Spin Bonus consists of two parts:

- Implementing the Free Spin creating logic.
- Adding new methods for sending Free Spin win and Free Spin bets, if bets are mandatory on the provider side.

10.1 Free Spin Creating

The provider should implement a method for creating Free Spin according to the model below and provide the endpoint.

The HTTP POST method is used for Free Spin creating request. The Content-Type of request is application/json. The following parameters are general for this request.

- playerId – The ID of the player, string
- operatorCode – The ID of the operator, string
- currency – The currency of the player, string
- externalReferenceId – The unique identifier of the request, string
- freeRoundValidity – The date before which the player can receive the Free Spins, datetime string, with “dd-MM-yyyy HH:MM:ss” format
- bonusMoneyValidity – The date before which the player can use the Free Spins, datetime string, with “dd-MM-yyyy HH:MM:ss” format
- numberOfFreeRounds – Count of given Free Spins, string
- gamelds – Ids of games for which the Free Spins are given, string
- wagerRequirement – The value of wagering factor. If the provider does not support it, then the value will be 0, int.
- automaticForfeitValue – Minimal bonus money value, after which the Free Spin will be canceled. This is active only when the wagering is enabled, decimal

Response parameter:

- result – request result (true if success and false if failed)
- err_desc – error description (if “result” is false)
- err_code – error code (if “result” is false)
- referenceId – unique identifier of creating request on provider side (string)