

# **b2b slots INTEGRATION**

# Table of contents

1. The Protocol.....	3
2. Creating Backend Communication.....	3
3. Integration API's .....	4
3.1 Open Game URL.....	5
3.2 Authentication of User API.....	6
3.3 Debit API.....	7
3.4 Credit API.....	8
3.5 Rollback API.....	9
3.6 Get Features API.....	11
3.7 Activate Features API.....	12
3.8 End Features API.....	14
4 Dictionaries.....	16
4.1 Currencies Dictionary.....	16
4.2 Game Codes.....	17
4.2 Error Codes.....	18
5. Examples.....	19
5.1 Auth.....	19
5.2 Debit.....	20
5.3 Credit.....	21
5.4 Rollback.....	22
5.5 Get Features.....	23
5.6 Activate Features.....	24
5.7 End Features.....	25

## 1. The Protocol

The integration uses HTTPS POST communications based using Json message type. The operator can use only HTTPS to communicate with our servers.

For any synchronous or asynchronous communication regarding financial operations such as debit, credit, and rollback, the Integrator backend must receive a response with the same transaction\_id parameter as the parameter sent in the request.

Json format is used for the backend communication for all kind of APIs

## 2. Creating Backend Communication

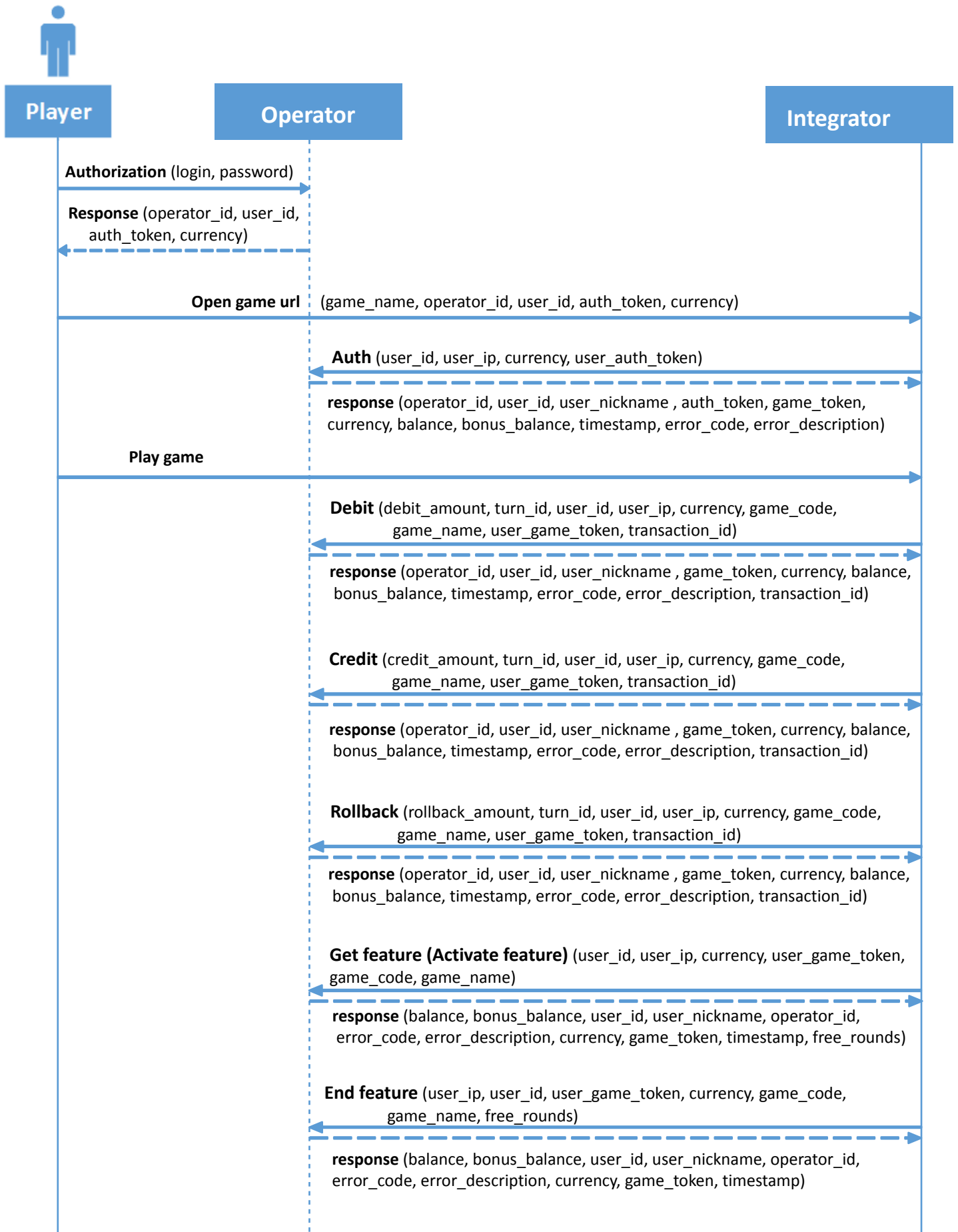
In order to create backend calls it is necessary to retrieve the URL, Communication Type (protocol name), and exact names of back-end operations.

What follows is the list of possible backend operations:

- auth
- debit
- credit
- rollback (debit refund - in the case of exceptions)
- get features
- end features

### 3. Integration API's

The following sequence diagram describes the communication between the Player, Integrator and the operator for every call to open the games.



- Players begin authorization on the operator's side and should receive the following parameters to use in **Game URL**:

Parameter name	Data Type/ Description
operator_id	int (11)
	Defines the particular operator that the player belongs to.
user_id	string (128)
	An operator's unique identifier for each player
auth_token	string (255)
	An unique identifier which is generated and sent from the operator in order to identify an interaction session
currency	string
	The currency code for the player's balance in a particular session

### 3.1 Open Game URL

The following table describes the parameters supplied by the operator to the Integrator required to launch the game.

```
https://int.apiforb2b.com/games/<game_name>.game?operator_id=<operator_id>
&user_id=<user_id>&auth_token=<auth_token>&currency=<currency>
```

Parameter name	Data Type/ Description
game_name	string
	The unique name of the game that the player wants to launch
operator_id	int (11)
	Defines the particular operator that the player belongs to.
user_id	string (128)
	An operator's unique identifier for each player
auth_token	string (255)
	An unique identifier which is generated and sent from the operator, in order to identify an interaction session
currency	string
	The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)

Example:

```
https://int.apiforb2b.com/games/BookOfRaClassic.game?
operator_id=0&user_id=1&auth_token=2a9a36b3487129aa899a57b161325ac6e3772174&currency=BTC
```

## 3.2 Authentication of User API

- Description**

The Authentication API enables the Integrator system to verify the information received in the launch protocol. Only once the operators approves this information will the game be available to the player.

Parameter name		Data Type/ Description
<b>Input parameters</b>		
<b>api</b>		String, value = "do-auth-user-ingame"
		An unique api name
<b>data</b>		JsonObject
		The json object of parameters below
<b>user_id</b>		string (128)
		An operator's unique identifier for each player
<b>user_ip</b>		string (30)
		The Internet IP of the computer on which the end user is playing
<b>user_auth_token</b>		string (255)
		An unique identifier which is generated and sent from the operator, in order to identify an interaction session
<b>currency</b>		string
		The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (will be expanded in the future on request)
<b>Output parameters</b>		
<b>api</b>		String
		An unique api name that sent from Integrator in request must be set to value is "do-auth-user-ingame"
<b>success</b>		Boolean
		The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
	<b>operator_id</b>	Int (11) The operator's unique identifier
<b>user_id</b>		String (128)
		An operator's unique identifier for each player.
<b>user_nickname</b>		String (32)
		The player's nickname. Characters and numbers only.
<b>balance</b>		String (20)
		Money available in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
<b>bonus_balance</b>		String (20)
		Available bonus amount in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
<b>auth_token</b>		String (255)
		The player's initial user_auth_token from operator (by launching the game url) - we use this for validation.
<b>game_token</b>		String (255)
		Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
<b>error_code</b>		int
		Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
<b>error_description</b>		String (255)
		A string that describes the response. This string is not the message to the player but rather displays details of the error.

	<b>currency</b>	String The currency the balance is in.
	<b>timestamp</b>	String (15) Time representation in UNIX milliseconds format.

### 3.3 Debit API

- Description**

The Debit API enables the Integrator platform to withdraw money from the player's wallet through the operator's server.

Parameter name		Data Type/ Description
<b>Input parameters</b>		
<b>api</b>		String, value = "do-debit-user-ingame" An unique api name
<b>data</b>		JsonObject The json object of parameters below
	<b>user_id</b>	string (128) An operator's unique identifier for each player
	<b>user_ip</b>	string (30) The Internet IP of the computer on which the end user is playing
	<b>user_game_token</b>	string (255) An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.
	<b>currency</b>	string The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
	<b>turn_id</b>	BigInt An unique key used to identify a game cycle.
	<b>transaction_id</b>	String (36) An unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once.
	<b>game_code</b>	Int An unique game code in the Integrator
	<b>game_name</b>	String An unique game name
	<b>debit_amount</b>	String (20) The amount requested to be withdrawn from the player's wallet. The value must be a positive number above zero. String representation of BigDecimal(15,5)
<b>Output parameters</b>		
<b>api</b>		String The unique api name that sent from the Integrator in request, value must be set to "do-debit-user-ingame"
<b>success</b>		Boolean The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
	<b>operator_id</b>	Int (11) The operator's unique identifier
	<b>transaction_id</b>	String (36) The same transaction_id as was sent in the request for this response. An unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.
	<b>user_id</b>	String (128) An operator's unique identifier for each player.

	<b>user_nickname</b>	String (32) The player's nickname. Characters and numbers only.
	<b>balance</b>	String (20) Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_balance</b>	String (20) Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_amount</b>	String (20) The bonus amount from the total debit amount. * Cannot be more than debit amount. String representation of BigDecimal(15,5)
	<b>game_token</b>	String (255) Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255) A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String The currency of the balance
	<b>timestamp</b>	String (15) Time representation in UNIX milliseconds format.

### 3.4 Credit API

- Description**

The Credit API enables the Integrator platform to perform a financial transaction of returning money into the player's wallet through the operator's server. When calculating the results, a player that loses has a credit amount of 0. A player that wins, is credited the total of the payout amount.

Parameter name	Data Type/ Description
<b>Input parameters</b>	
<b>api</b>	String, value = "do-credit-user-ingame" An unique api name
<b>data</b>	JsonObject The json object of parameters below
<b>user_id</b>	string (128) An operator's unique identifier for each player
<b>user_ip</b>	string (30) The Internet IP of the computer on which the end user is playing
<b>user_game_token</b>	string (255) An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.
<b>currency</b>	string The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
<b>turn_id</b>	BigInt An unique key, to identify a game cycle.
<b>transaction_id</b>	String (36) An unique key used to indicate a specific financial activity. This key will guarantee that a transaction is handled only once.



	<b>game_code</b>	Int An unique game code in the Integrator
	<b>game_name</b>	String An unique game name
	<b>credit_amount</b>	String(20) The amount requested to be fill up the player's wallet. The value must be a positive number above zero. String representation of BigDecimal(15,5)
<b>Output parameters</b>		
<b>api</b>		String An unique api name that sent from the Integrator in request, value must be set to "do-credit-user-ingame"
<b>success</b>		Boolean The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
	<b>operator_id</b>	Int (11) The operator's unique identifier
	<b>transaction_id</b>	String (36) The same transaction_id as was sent in the request for this response. An unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.
	<b>user_id</b>	String (128) An operator's unique identifier for each player.
	<b>user_nickname</b>	String (32) The player's nickname. Characters and numbers only.
	<b>balance</b>	String (20) Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_balance</b>	String (20) Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_amount</b>	String(20) The bonus amount from the total credit amount. * Cannot be more than credit amount. String representation of BigDecimal(15,5)
	<b>game_token</b>	String (255) Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255) A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String The currency of the balance
	<b>timestamp</b>	String (15) Time representation in UNIX milliseconds format.

## 3.5 Rollback API

- **Description**

During a debit transaction, there is a possibility of transactions being canceled and therefore rolled back. Rollback API is designed to deal with exceptions received from the operator (i.e. when the server disconnects, or an invalid error response is returned from the operator's end). The debit

transaction is converted into a rollback transaction, using the same transaction id and the debit amount field is renamed to rollback amount. It is crucial for the rollback to use a separate method from the debit in order to avoid errors and data mix-ups.

Parameter name		Data Type/ Description
<b>Input parameters</b>		
<b>api</b>		String, value = "do-rollback-user-ingame"
		An unique api name
<b>data</b>		JsonObject
		The json object of parameters below
<b>user_id</b>		string (128)
		An operator's unique identifier for each player
<b>user_ip</b>		string (30)
		The Internet IP of the computer on which the end user is playing
<b>user_game_token</b>		string (255)
		An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.
<b>currency</b>		string
		The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
<b>turn_id</b>		BigInt
		An unique key, to identify a game cycle.
<b>transaction_id</b>		String (36)
		An unique key to indicate a specific financial activity. This key will guarantee that a transaction is handled only once.
<b>game_code</b>		Int
		An unique game code in the Integrator
<b>game_name</b>		String
		An unique game name
<b>rollback_amount</b>		String (20)
		The amount requested to be returned to the player's wallet. The value must be a positive number above zero. String representation of BigDecimal(15,5)
<b>Output parameters</b>		
<b>api</b>		String
		An unique api name that sent from the Integrator in request, value must be set to "do-rollback-user-ingame"
<b>success</b>		Boolean
		The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
	<b>operator_id</b>	Int (11) The operator's unique identifier
<b>transaction_id</b>		String (36)
		The same transaction_id as was sent in the request for this response. An unique key to indicate a specific financial transaction. This key will guarantee that a transaction is handled only once.
<b>user_id</b>		String (128)
		An operator's unique identifier for each player.
<b>user_nickname</b>		String (32)
		The player's nickname. Characters and numbers only.
<b>balance</b>		String (20)
		Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
<b>bonus_balance</b>		String (20)
		Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)

	<b>game_token</b>	String (255) Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255) A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String The currency of the balance
	<b>timestamp</b>	String (15) Time representation in UNIX milliseconds format.

### 3.6 Get Features API

- Description**

The Get Features API enables the Integrator platform to request any kind of bonus program like free rounds using the operator's side.

Parameter name		Data Type/ Description
<b>Input parameters</b>		
<b>api</b>		String, value = "do-get-features-user-ingame" An unique api name
<b>data</b>		JsonObject The json object of parameters below
	<b>user_id</b>	string (128) An operator's unique identifier for each player
	<b>user_ip</b>	string (30) The Internet IP of the computer on which the end user is playing
	<b>user_game_token</b>	string (255) An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.
	<b>currency</b>	string The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
	<b>game_code</b>	Int An unique game code in the Integrator
<b>Output parameters</b>		
<b>api</b>		String An unique api name that sent from the Integrator in request, value must be set to "do-get-features-user-ingame"
<b>success</b>		Boolean The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
	<b>operator_id</b>	Int (11) The operator's unique identifier
	<b>user_id</b>	String (128) An operator's unique identifier for each player.
	<b>user_nickname</b>	String (32) The player's nickname. Characters and numbers only.

	<b>balance</b>	String (20)	Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_balance</b>	String (20)	Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>game_token</b>	String (255)	Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int	Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255)	A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String	The currency of the balance
	<b>timestamp</b>	String (15)	Time representation in UNIX milliseconds format.
	<b>free_rounds</b>	JsonObject	The Json Object parameters below
		<b>id</b>	int An unique identifier of feature
		<b>count</b>	int Count of free games
		<b>bet</b>	int Bet to be set to execute features
		<b>lines</b>	int Lines to be set to execute features
		<b>mpl</b>	int Multiplier of feature games
		<b>cp</b>	double Credit price

### 3.7 Activate Features API

- Description**

The Activate Features API enables the Integrator platform to notify operator about player's activated bonus.

Parameter name	Data Type/ Description
<b>Input parameters</b>	
<b>api</b>	String, value = "do-activate-features-user-ingame" The unique api name
<b>data</b>	JsonObject The json object of parameters below
	<b>user_id</b>
	string (128) An operator's unique identifier for each player
	<b>user_ip</b>
	string (30) The Internet IP of the computer on which the end user is playing
	<b>user_game_token</b>
	string (255) An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.

	<b>currency</b>	string
		The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
	<b>game_code</b>	Int
		An unique game code in the Integrator
	<b>free_rounds</b>	JsonObject
		Json Object of parameters below
	<b>id</b>	int
		An unique feature id
	<b>win</b>	double
		Win amount in feature. The amount requested to be fillup to the player's wallet. The value must be a positive number above or equal zero
<b>Output parameters</b>		
	<b>api</b>	String
		The unique api name that sent from the Integrator in request, value must be set to "do-activate-features-user-ingame"
	<b>success</b>	Boolean
		The result of the api request, value must always be true
	<b>answer</b>	JsonObject of parameters below
	<b>operator_id</b>	Int (11)
		The operator's unique identifier
	<b>user_id</b>	String (128)
		An operator's unique identifier for each player.
	<b>user_nickname</b>	String (32)
		The player's nickname. Characters and numbers only.
	<b>balance</b>	String (20)
		Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>bonus_balance</b>	String (20)
		Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
	<b>game_token</b>	String (255)
		Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int
		Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255)
		A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String
		The currency of the balance
	<b>timestamp</b>	String (15)
		Time representation in UNIX milliseconds format.

## 3.8 End Features API

- **Description**

The End Features API enables the Integrator platform come up with a notification regarding the end of previous features activated by Get Features through the operator's side.

Parameter name		Data Type/ Description
<b>Input parameters</b>		
<b>api</b>		String, value = "do-end-features-user-ingame"
		The unique api name
<b>data</b>		JsonObject
		The json object of parameters below
<b>user_id</b>		string (128)
		An operator's unique identifier for each player
<b>user_ip</b>		string (30)
		The Internet IP of the computer on which the end user is playing
<b>user_game_token</b>		string (255)
		An unique identifier which is generated before (in auth api) by the operator to identify a session's interactions.
<b>currency</b>		string
		The currency code for the player's balance in a particular session: BTC, LTC, USD, EUR, RUB, KZT, UAH (can be expanded in the future)
<b>game_code</b>		Int
		An unique game code in the Integrator
<b>free_rounds</b>		JsonObject
		Json Object of parameters below
<b>id</b>		int
		An unique feature id
<b>win</b>		double
		Win amount in feature. The amount requested to be credited to the player's wallet. The value must be a positive number above or equal zero
<b>Output parameters</b>		
<b>api</b>		String
		The unique api name that sent from the Integrator in request, value must be set to "do-end-features-user-ingame"
<b>success</b>		Boolean
		The result of the api request, value must always be true
<b>answer</b>		JsonObject of parameters below
<b>operator_id</b>		Int (11)
		The operator's unique identifier
<b>user_id</b>		String (128)
		An operator's unique identifier for each player.
<b>user_nickname</b>		String (32)
		The player's nickname. Characters and numbers only.
<b>balance</b>		String (20)
		Available money in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
<b>bonus_balance</b>		String (20)
		Available bonus balance in the player's wallet at the time of the request. The value will be presented in the currency of that session. String representation of BigDecimal(15,5)
		String (255)

	<b>game_token</b>	Not to be confused with the input auth_token. This token is the session token- An unique identifier which is generated by the operator to identify a session's interactions.
	<b>error_code</b>	int Identifies whether or not the request has been processed successfully. If no error is detected, this code value is 0
	<b>error_description</b>	String (255) A string that describes the response. This string is not the message to the player but rather gives details of the error.
	<b>currency</b>	String The currency of the balance
	<b>timestamp</b>	String (15) Time representation in UNIX milliseconds format.

## 4 Dictionaries

### 4.1 Currencies Dictionary

Currency code	Currency name
BTC	Bitcoin cryptocurrency
LTC	Litecoin cryptocurrency
USD	United States Dollar
EUR	Euro
RUB	Russian Ruble
KZT	Kazakhstan tenge
UAH	Ukrainian hryvnia
*	*

\* Can be added any currency in future



## 4.2 Game Codes

The full list \* of games can be obtained in json format by executing a request using the GET method, specifying the operator id in the parameter operator\_id:

```
https://int.apiforb2b.com/frontendsrv/apihandler.api?cmd={%22api%22:%22ls-games-by-operator-id-get%22,%22operator_id%22:%220%22}
```

Getting the game image from JSON:

```
{
  "ico_baseurl": "/game/icons/",
  "groups": [
    {
      "games": [
        {
          ...
          "icons": [
            {
              "ic_h": 221,
              "ic_w": 300,
              "ic_name": "ReelKing_300_221.jpg"
            },
            ...
          ],
          "gm_title": "Reel King"
        },
        ...
      ],
      ...
    }
  ]
}
```

```
https://int.apiforb2b.com[ico_baseurl][ic_name]
```

Example:

```
https://int.apiforb2b.com/game/icons/ReelKing_300_221.jpg
```

\* New games always in developing. This list can be extended at any time

### 4.3 Error Codes

error_code	error_description
0	Completed successfully
1	General error
2	*
3	Insufficient funds
4	Token not found
5	User not found
6	User blocked
7	Transaction not found
8	Transaction timed out
9	*
10	*
11	*

\* - Reserved codes for future use

Note: In the event that the same transaction\_id is sent more than once, you must send us the same response with the original error code. In the error description write- transaction has already been proceeded.

# 5. Examples

## 5.1 Auth

### *Integrator Request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_auth_token": "638b779ea073609b28374364bf1eb0de488fb9b5",
    "currency": "BTC"
  },
  "api": "do-auth-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 0,
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.67713",
    "bonus_balance": "0.00",
    "auth_token": "638b779ea073609b28374364bf1eb0de488fb9b5",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "timestamp": "1478008017191"
  },
  "success": true,
  "api": "do-auth-user-ingame"
}
```

## 5.2 Debit

### *Integrator request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "turn_id": 4,
    "transaction_id": "2fa7bfb2-a03a-11e6-8347-00ffa41930b3",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
    "debit_amount": "0.02"
  },
  "api": "do-debit-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 0,
    "transaction_id": "2fa7bfb2-a03a-11e6-8347-00ffa41930b3",
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.67713",
    "bonus_balance": "0.0",
    "bonus_amount": "0.0",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "timestamp": "1478008244952"
  },
  "success": true,
  "api": "do-debit-user-ingame"
}
```

## 5.3 Credit

### *Integrator Request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "turn_id": 5,
    "transaction_id": "2fa7bfb2-a03a-11e6-8347-00ffa41930b3",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
    "credit_amount": "0.02"
  },
  "api": "do-credit-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 0,
    "transaction_id": "2fb1fd48-a03a-11e6-8347-00ffa41930b3",
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.67913",
    "bonus_balance": "0.00",
    "bonus_amount": "0.00",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "timestamp": "1478008244987"
  },
  "success": true,
  "api": "do-credit-user-ingame"
}
```

## 5.4 Rollback

### *Integrator Request*

```
{
  "data": {
    "user_ip": "127.0.0.1",
    "user_id": "1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "turn_id": 5,
    "transaction_id": "2fa7bfb2-a03a-11e6-8347-00ffa41930b3",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
    "rollback_amount": "0.02"
  },
  "api": "do-rollback-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 0,
    "transaction_id": "2fb1fd48-a03a-11e6-8347-00ffa41930b3",
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.67913",
    "bonus_balance": "0.00",
    "bonus_amount": "0.00",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "timestamp": "1478008244987"
  },
  "success": true,
  "api": "do-rollback-user-ingame"
}
```

## 5.5 Get Features

### *Integrator Request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
  },
  "api": "do-get-features-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 1,
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.67913",
    "bonus_balance": "0.00",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "timestamp": "1478008244987",
    "free_rounds": {
      "id": 1,
      "count": 15,
      "bet": 5,
      "lines": 10,
      "mpl": 2,
      "cp": "1.00"
    }
  },
  "success": true,
  "api": "do-get-features-user-ingame"
}
```

## 5.6 Activate Features

### *Integrator Request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
    "free_rounds": {
      "id": 1
    }
  },
  "api": "do-activate-features-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 0,
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.6913",
    "bonus_balance": "0.00",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "timestamp": "1478008244987"
  },
  "success": true,
  "api": "do-activate-features-user-ingame"
}
```



## 5.7 End Features

### *Integrator Request*

```
{
  "data": {
    "user_id": "1",
    "user_ip": "127.0.0.1",
    "user_game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "currency": "BTC",
    "game_code": 702,
    "game_name": "BookOfRaClassic",
    "free_rounds": {
      "id": 1,
      "win": 1000
    }
  },
  "api": "do-end-features-user-ingame"
}
```

### *Operator Response*

```
{
  "answer": {
    "operator_id": 1,
    "user_id": "1",
    "user_nickname": "Anonymous",
    "balance": "855.6913",
    "bonus_balance": "0.00",
    "error_code": 0,
    "error_description": "ok",
    "currency": "BTC",
    "game_token": "c3720505ad4dfd2b34cdb55bc524f3ffa6941ad7",
    "timestamp": "1478008244987"
  },
  "success": true,
  "api": "do-end-features-user-ingame"
}
```